Error Detecting Dual Basis Bit Parallel Systolic Multiplication Architecture over GF(2^m)

Ashutosh Kumar Singh, Asish Bera, Hafizur Rahaman, Jimson Mathew, and Dhiraj K. Pradhan

***Abstract*—An error tolerant hardware efficient very large scale integration (VLSI) architecture for bit parallel systolic multiplication over dual base, which can be pipelined, is presented. Since this architecture has the features of regularity, modularity and unidirectional data flow, this structure is well suited to VLSI implementations. The length of the largest delay path and area of this architecture are less compared to the bit parallel systolic multiplication architectures reported earlier. The architecture is implemented using Austria Micro System's 0.35 μm CMOS (complementary metal oxide semiconductor) technology. This architecture can also operate over both the dual-base and polynomial base.**

***Index Terms*—Bit parallel, error correction, finite field, Reed-Solomon (RS) codes, systolic, very large scale integration (VLSI) testing.**

1. Introduction

Finite field also known as Galois Field arithmetic operations over GF(2^m) finds increasing applications in public-key cryptography, error detecting and correcting code[1], VLSI (very large scale integration) testing[2], digital signal processing[3]. There are different equivalent representations of the elements of the finite field over GF(2^m), e.g. polynomial base (PB), normal base, and dual base. Dual-basis operators frequently have the lowest hardware requirements of all available operators[4],[5]. Two basic operations over GF(2^m) are addition and multiplication. Addition over GF(2^m) is relatively straightforward to implement, requiring at most m XOR gates. Multiplication operation is much more expensive in terms of gate count and clock cycle. Other operations of the GF(2^m) fields like exponentiation, division, and inversion can be performed by repeated multiplications. Based on different base representation, a variety of architectures for multiplication have been proposed. For high speed VLSI implementation, the preferred multiplier architecture is systolic array architecture. In this type of architecture, a basic cell is repeated in an array and signals flow unilaterally between neighbours. Polynomial basis (PB) systolic array multipliers in GF(2^m) can be classified into four categories, namely bit serial[6], bit-parallel, hybrid and digit-serial[7]. The bit serial architecture has minimum area and minimum throughput among all the categories. The problem with serial architecture is its latency. The bit-serial architecture, which processes one bit of input data per clock cycle, is area-efficient and suitable for low-speed applications.

The most widely used bit serial multiplier is dual basis Berlekamp bit serial multiplier[8]. This multiplier requires less hardware. PB bit-serial and bit-parallel systolic multipliers were presented in [9] and [10]. A bit-serial dual basis systolic multiplier over GF(2^m) was presented in [11], which requires higher hardware compared to that needed for multiplier proposed in [12] and does not support pipelining. To support pipelining, a modified version which requires less hardware is presented in [13]. The bit parallel multiplier needs largest area and provides maximum throughput. Bit-parallel architecture, capable of processing one whole word of input data per clock cycle, is ideal for high-speed applications when pipelined at the bit-level. These architectures are typical examples of the area-speed tradeoff paradigm. Mastrovito has proposed an algorithm along with its hardware architecture for PB multiplication[14] known as the Mastrovito algorithm/multiplier. A formulation for polynomial basis multiplication and generalized bit-parallel hardware architecture for special reduction polynomials has been presented in [15]. A testable polynomial basis bit parallel multiplier circuits over GF(2^m) was presented in [16]. Although bit-serial dual basis multipliers have been widely employed in applications such as Reed-Solomon (RS) encoders[11],[17], it was proven in [5] that it is advantageous of employing bit-parallel dual basis multipliers, particularly in more

Manuscript presented at 2009 IEEE Circuits and Systems International Conference on Testing and Diagnosis, April 28-29, 2009; received September 24, 2009.

A. K. Singh is with CS Dept., School of Engineering, Curtin University of Technology, Malaysia (e-mail: ashutosh.s@curtin.edu.my).

A. Bera is with School of VLSI Technology, Bengal Engg. & Sc. University, Shibpur, India.

H. Rahaman is with Dept. of Information Technology, Bengal Engg. & Sc. University, Shibpur, India. He is currently visiting University of Bristol, UK. (e-mail: hafizur@cs.bris.ac.uk, rahaman_h@hotmail.com).

J. Mathew and D. K. Pradhan are with Computer Science Dept., University of Bristol, UK, (e-mail: jimson @cs.bris.ac.uk, pradhan@ cs.bris.ac.uk).

complex circuits such as RS decoders and syndrome calculators. Bit-parallel dual basis multipliers therefore provide reduced complexity constant multipliers. In this paper, we present a hardware efficient fast bit parallel systolic architecture with error detecting capability using parity prediction technique over dual base which can be pipelined.

The rest of the paper is organized as follows. Section 2 briefly describes the preliminaries. In section 3, we propose systolic bit-parallel and digit serial architecture based on MM algorithm. Section 4 presents analysis and discussion on these architectures. The experimental results have appeared in Section 5. Finally, we conclude our discussions in Section 6.

2. Preliminaries

2.1 Polynomial Multiplication

Let GF(N) denote a set of N elements, where N is a power of a prime number, with two special elements 0 and 1 representing the additive and multiplicative identities respectively and two operator addition '+' and multiplication '·'. The GF(N) defines a finite field, if it forms a commutative ring with identity over these two operators in which every element has a multiplicative inverse. Finite fields can be generated with primitive polynomials of the form $P(x)=x^{m-1}+\sum_{i=0}^{m} p_i x^i$, where $p_i \in \mathrm{GF}(2)$ [1]. It is conventional to represent the elements of GF(2^m) as a power of the primitive element α, where α is the root of $P(x)$, i.e., $P(\alpha)=0$. The set $\{1, \alpha, \ldots, \alpha^{m-1}\}$ is referred to as polynomial basis or standard basis. Each element $A \in \mathrm{GF}(2^m)$ can be expressed with respect to the PB as a polynomial of degree m over GF(2), i.e., $A(x)=\sum_{i=0}^{m-1} a_i x^i$ where $a_i \in \mathrm{GF}(2)$. Given $A, B \in \mathrm{GF}(2^m)$, the PB multiplication over GF(2^m) can be defined as $C(x)=A(x)B(x) \bmod P(x)$. In practice, $C(x)$ is obtained in two steps: polynomial multiplication and modulo reduction.

2.2 Dual Basis Multiplication

Let F_p^m denote the set of all linear function f: GF(p^m)→GF(p). A well known linear function is the trace function which is frequently used to produce the finite field multipliers. There are number of other linear functions including trace functions. Here, we use the definition of the duality of two bases[13],[14] as given below.

Definition. Let $\{\lambda_i\}$ and $\{\mu_i\}$ be bases for GF(2^m), let f: GF(2^m)→GF(2) be a linear function and let $\beta \in \mathrm{GF}(2^m)$, $\beta \neq 0$. Then the bases are said to be dual with respect to f and β if

$$f\left(\beta\lambda_i\mu_j\right)=\begin{cases}1, & \text{if } i=j\\ 0, & \text{if } i\neq j.\end{cases}$$

In this case $\{\lambda_i\}$ is the standard basis and $\{\mu_i\}$ is the dual basis. We now restate the multiplication algorithm utilized here. This result was first presented in the context of division[14] but has subsequently been used to describe finite-field multiplication[18]. Furthermore, as observed in [19], the following represents a generalized and alternative representation of Berlekamp bit-serial multiplier.

Theorem 1[13]. Let a, b, $c \in \mathrm{GF}(p^m)$ such that $c=ab$. Further, let α be a root of the defining irreducible polynomial for the field, let $\beta \in \mathrm{GF}(2^m)$, $f \in F_2^m$ and represent c over the polynomial basis by $a=\sum_{i=0}^{m-1} a_i\alpha^i$, then the following relation holds:

$$\begin{bmatrix} f(b\beta) & f(b\beta\alpha) & \cdots & f(b\beta\alpha^{m-1}) \\ f(b\beta\alpha) & f(b\beta\alpha^2) & \cdots & f(b\beta\alpha^{m}) \\ \vdots & \vdots & \vdots & \vdots \\ f(b\beta\alpha^{m-1}) & f(b\beta\alpha^{m}) & \cdots & f(b\beta\alpha^{2m-2}) \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \end{bmatrix} = \begin{bmatrix} f(c\beta) \\ f(c\beta\alpha) \\ \vdots \\ f(c\beta\alpha^{m-1}) \end{bmatrix}. \quad (1)$$

We have modified (1) as follows:

$$\begin{bmatrix} b_0 & b_1 & \ldots & b_{m-1} \\ b_1 & b_2 & \ldots & b_m \\ \vdots & \vdots & \vdots & \vdots \\ b_{m-1} & b_m & \ldots & b_{2m-2} \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix} \quad (2)$$

where $b_k=f(b\beta\alpha^k)$ (k=0, 1,…, 2m–2) and $c_k=f(c\beta\alpha^k)$ (k=0, 1, …, m–1). If f and β are taken as in the preceding definition, c_k and b_k, (k=0, 1, …, m–1) in (1) are the dual-basis coefficients of c and b, respectively. Thus to make use of (1) in a systolic multiplier, one must first generate the values of b_k (k=m, m + 1, …, 2m–2).

If $p(x)=\sum_{i=0}^{m-1} p_i x^i + x_m$ is the defining irreducible polynomial for the field, then

$$b_m = f\left(b\beta\alpha^m\right) = f\left(b\left(\beta\sum_{j=0}^{m-1} p_j a^j\right)\right)$$

$$=\sum_{j=0}^{m-1} p_j f\left(b\beta\alpha^j\right)=\sum_{j=0}^{m-1} p_j b_j$$

and then

$$b_{m+k} = f\left(b\beta\alpha^{m+k}\right) = f\left(b\left(\beta\sum_{j=0}^{m-1} p_j\alpha^{j+k}\right)\right)$$

$$=\sum_{j=0}^{m-1} p_j f\left(b\beta\alpha^{j+k}\right) = \sum_{j=0}^{m-1} p_j b_{j+k}.$$

Then in general

$$b_{m+k} = \sum_{j=0}^{m-1} p_j b_{j+k} \tag{3}$$

where b_k (k=0, 1, …, m–1) are the dual basis coefficients of b and α is root of $p(x)$. After computing the values of b_k from (2), we need to carry out the matrix multiplication given in (1). Now we consider the implementation of this multiplication algorithm in the design of a bit-parallel systolic multiplier.

3. Bit Parallel Dual Basis Multiplier

3.1 Proposed Architecture

Let a, b, $c\in$GF(2^m) such that $c=ab$ and let $\{\mu_i\}$ be the dual basis to the polynomial basis for $\beta\in$GF(2^m) and $f \in F_2^m$. Representing b over the dual basis by $b = \sum_{i=0}^{m-1} b_i\mu_i$ and a over the polynomial basis by a, $a = \sum_{i=0}^{m-1} a_i\alpha^i$. We can derive following equation from (2):

$$c_0=b_0a_0+b_1a_1+\ldots+b_{m-1}a_{m-1}$$

$$c_1=b_1a_0+b_2a_1+\ldots+b_ma_{m-1};\ \ldots$$

$$c_{m-1}=b_{m-1}a_0+b_ma_1+\ldots\ +b_{2m-2}\,a_{m-1}$$

where b_{m+k} ($k\geq$0) are given by (3). From these equations, it can be seen that m product bits are generated by m identical functions of the form.

$$h(b,\ a)=b_ka_0+b_{k+1}a_1+\ldots+\ b_{k+\ m-1}\ a_{m-1}. \tag{4}$$

A bit-parallel dual basis multiplier over GF(2^m) can, therefore, be constructed using two cells. We introduce cell-1 as shown in Fig. 2 to generate (3) and also introduce a cell-2 for generating (2) as shown in Fig. 1. An example of such a multiplier over GF(2^4) is given below.

Example 1. Let $p(x)=x^4+x+1$ be the defining irreducible polynomial and let a be a root of $p(x)$. From (4), we can write as follows:

$$h(b,\ a)=b_ka_0+b_{k+1}a_1+b_{k+2}a_2+b_{k+3}a_3. \tag{5}$$

This equation can be implemented by the circuit as shown in Fig. 2. From $p(x)=x^4+x+1$ and (3) and (4), we can derive the values of b_4, b_5, b_6 as follows:

$$b_4=b_1+b_0,\quad b_5=b_2+b_1,\quad b_6=b_3+\ b_2.$$

Equation (2) for this example is given below.

$$\begin{bmatrix} b_0 & b_1 & b_2 & b_3 \\ b_1 & b_2 & b_3 & b_4 \\ b_2 & b_3 & b_4 & b_5 \\ b_3 & b_4 & b_5 & b_6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

The m^2 cells of Fig. 1 and m cells of Fig. 2 are then combined to form the full bit-parallel dual basis multiplier for GF(2^4) as shown in Fig. 3. If $b = \sum_{i=0}^{m-1} b_i\mu_i$ is the dual basis representation of b and $a = \sum_{i=0}^{m-1} a_i\alpha^i$ is the polynomial basis representation of a, the product bits c_i (i= 0, 1, 2, 3) become available on the output lines. In the architecture, b_4, b_5 and b_6 are generated by the block diagram of Fig. 2. In general, Fig. 2 represents the sum of partial products (2), i.e., $b_{m+k} = \sum_{j=0}^{m-1} p_j b_{j+k}$, k=0, 1, …, m–2. The partial sum in the matrix multiplication in (1) is generated by the block diagram of Fig. 1.

In BP Systolic dual basis multiplier design of [13], there exist two datapaths, one is horizontal and the other is vertical. The vertical datapath generates partial sum in matrix multiplication of (1). The horizontal data path generates partial sum of (2). There is a bottleneck to support pipelining in this design. The horizontal data path consists of AND-XOR binary tree, the depth of tree is $O(m)$. We try to modify the horizontal data path by replacing the binary tree of depth $O(m)$ with a binary tree of depth of $O(\log_2)$. For this purpose, we introduce a new cell (see Fig. 2) to generate (2). The complete circuit for dual basis systolic multiplier over GF(2^4) is shown in Fig. 3. Latches are introduced in Fig. 3, to make this architecture suitable for pipelining. There is m-clock cycle delay between b, c entering in the multiplier and becoming available in the output lines. After the initial delay, results can be produced continuously one per clock cycle.

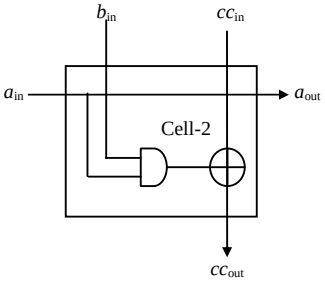

Fig. 1. Generation of partial products of (1).

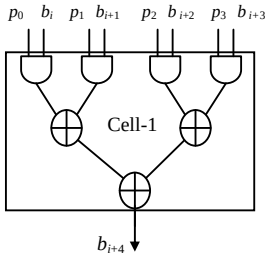

Fig. 2. Generation of the sum of partial products of (2).

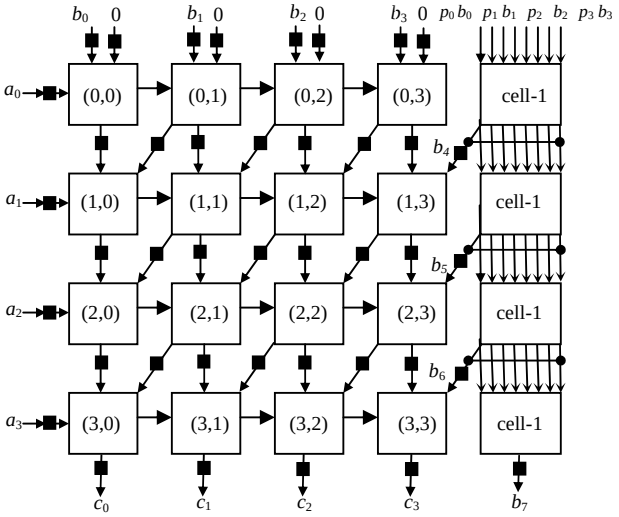

Fig. 3. Arrangement of systolic cells for bit-parallel multiplier for GF(2^4).

3.2 Hardware and Delay Analysis

We compare our proposed architecture with the bit parallel architecture described in [14]. Total hardware required for the architecture presented consists of m^2 cells. Each cell consists of two 2 input AND gates and two 2 input EXOR gates. Total circuit consists of $2m^2$ AND gates and $2m^2$ EXOR gates. Our proposed design requires 2 cells. The first cell consists of one AND gate and one EXOR gate. The second cell consists of m AND gates and $(m–1)$ EXOR gates. For m bit multipliers, the proposed architecture consists of m^2 first cells and m second cells. Total $2m^2$ AND gates and $(2m^2–m)$ EXOR gates are required. Overall saving in hardware is m EXOR gate.

Let D_A be the delay through a two-input AND gate and D_x be the delay through a two-input XOR gate. The longest delay path is given in (6).

$$\text{Longest delay} = \{mD_A + (\log_2 m + m - 1)D_X\}. \quad (6)$$

BP multiplier of [14] has a longest delay path of $\{(2m–1)[D_A+D_X]\}$, whereas the proposed multiplier has a longest delay path of $\{mD_A+(\log_2 m+m–1)D_X\}$. Hence, the proposed dual basis BP multiplier is hardware efficient and faster.

From Table 1, we can conclude that in this architecture, the number of AND gates are the same compared with previous architecture in [5], but for m-bit dual basis systolic multiplier m, the number of XOR gates are less required in this architecture as well as the longest path delay of this architecture is also reduced by m-bit for AND gates and for XOR gates delay is reduced by $\log_2 m$ instead of m.

Table 1: Hardware requirements and delays of dual basis bit parallel multiplier (DPM) presented in [14] and the proposed multiplier (DPM)

	DPM in [14]			Proposed DPM		
m	AND	XOR	Delay	AND	XOR	Delay
2	8	8	$3[D_A+D_X]$	8	6	$2D_A+2D_X$
3	18	18	$5[D_A+D_X]$	18	15	$3D_A+3.58D_X$
4	32	32	$7[D_A+D_X]$	32	28	$4D_A+5D_X$
5	50	50	$9[D_A+D_X]$	50	45	$5D_A+6.32D_X$
6	72	72	$11[D_A+D_X]$	72	66	$6D_A+7.58D_X$
7	98	98	$13[D_A+D_X]$	98	91	$7D_A+8.81D_X$
8	128	128	$15[D_A+D_X]$	128	120	$8D_A+10D_X$
9	162	162	$17[D_A+D_X]$	162	153	$9D_A+11.17D_X$
10	200	200	$19[D_A+D_X]$	200	190	$10D_A+12.32D_X$

Table 2: Comparison between two bit-parallel systolic multipliers

Properties		Reference [5]	Presented here
Number of cells		m^2	Cell 1: m^2 Cell 2: m
Circuit complexity	No. of 2 input AND gate	$2m^2$	$2m^2$
	No. of 2 input XOR gate	$2m^2$	$2m^2–m$
Largest delay path		$(2m–1)[D_A+D_X]$	$mD_A+(\log_2 m+m–1)D_X$

In Table 1, the hardware complexity and delays of the DPM in [5] and our proposed DPM architecture are given for GF(2^m) (m=2, 3, …, 10). From Table 2, it can be seen that for every case, the hardware complexity and delays of our proposed DPM architecture are less compared with those of the DPM architecture in [5].

4. Error Detection Using Parity Checking

We use error-detection scheme with a very high probability of detecting faults in the bit-parallel systolic multiplication over GF(2^m) using dual base with some additional outputs, called the check-bits as shown in Fig. 4. We assume that no interconnections or buses have any fault and each test phase with the test-circuits is separately controllable. At first, we attach parity-bits to the input elements b_p and a_p and multiplying (AND) the inputs we have:

$$b_p = b_0 \oplus b_1 \oplus b_2 \oplus b_3,\ a_p = a_0 \oplus a_1 \oplus a_2 \oplus a_3$$

$$\begin{aligned} b_p a_p &= (b_0 \oplus b_1 \oplus b_2 \oplus b_3)(a_0 \oplus a_1 \oplus a_2 \oplus a_3) \\ &= (b_0a_0 \oplus b_0a_1 \oplus b_0a_2 \oplus b_0a_3) \oplus (b_1a_0 \oplus b_1a_1 \oplus b_1a_2 \oplus b_1a_3) \\ &\quad \oplus (b_2a_0 \oplus b_2a_1 \oplus b_2a_2 \oplus b_2a_3) \oplus (b_3a_0 \oplus b_3a_1 \oplus b_3a_2 \oplus b_3a_3). \end{aligned}$$

From (2), we get

$$\begin{aligned} c_0 &= b_0a_0 \oplus b_1a_1 \oplus b_2a_2 \oplus b_3a_3 \\ c_1 &= b_1a_0 \oplus b_2a_1 \oplus b_3a_2 \oplus b_4a_3 \\ c_2 &= b_2a_0 \oplus b_3a_1 \oplus b_4a_2 \oplus b_5a_3 \\ c_3 &= b_3{}_0 \oplus b_4a_1 \oplus b_5a_2 \oplus b_6a_3. \end{aligned}$$

Now, we denote the modulo 2 addition of these outputs of the multiplier by

$$r = c_0 \oplus c_1 \oplus c_2 \oplus c_3.$$

Here, we add some extra lines and gates for the testing purposes which constitute the feedback lines y_i. Lines b_0, b_1, b_2, b_3 and some XOR and AND gates are used to produce the circuit suitable for the testing. Some lines are used as

feedback and are denoted by (y_1, y_2, y_3, y_4, y_5, y_6). So, some of the terms are eliminated when b_p, a_p are added by modulo 2 addition to form the parity check in the output line with the feedback lines.

The y_i lines are given as:

$$y_1= b_0a_1\oplus b_0a_2\oplus b_0a_3$$
$$y_2=b_1a_2\oplus b_1a_3$$
$$y_3=b_2a_3$$
$$y_4=b_4a_1\oplus b_5a_2\oplus b_6a_3$$
$$y_5=b_4a_2\oplus b_5a_3$$
$$y_6=b_4a_3.$$

The q line is derived from modulo addition of b_pc_p and the y_i lines.

$$q=b_pa_p\oplus y_1\oplus y_2\oplus y_3\oplus y_4\oplus y_5\oplus y=b_0a_0\oplus b_0a_1\oplus b_0a_2\oplus b_0a_3 \oplus b_1a_0\oplus b_1a_1\oplus b_1a_2\oplus b_1a_3\oplus b_2a_0\oplus b_2a_1\oplus b_2a_2\oplus b_2a_3 \oplus b_3a_0\oplus b_3a_1\oplus b_3a_2\oplus b_3a\oplus b_0a_1\oplus b_0a_2\oplus b_0a_3\oplus b_1a_2\oplus b_1a_3 \oplus b_2a_3\oplus b_4a_1\oplus b_5a_2\oplus b_6a_3\oplus b_4a_2\oplus b_5a\oplus b_4a_3=b_0a_0\oplus b_1a_0 \oplus b_1a_1\oplus b_2a_0\oplus b_2a_1\oplus b_2a_2\oplus b_3a_0\oplus b_3a_1\oplus b_3a_2\oplus b_3a_3\oplus b_4a_1 \oplus b_4a_2\oplus b_4a_3\oplus b_5a_2\oplus b_5a_3\oplus b_6a_3.$$

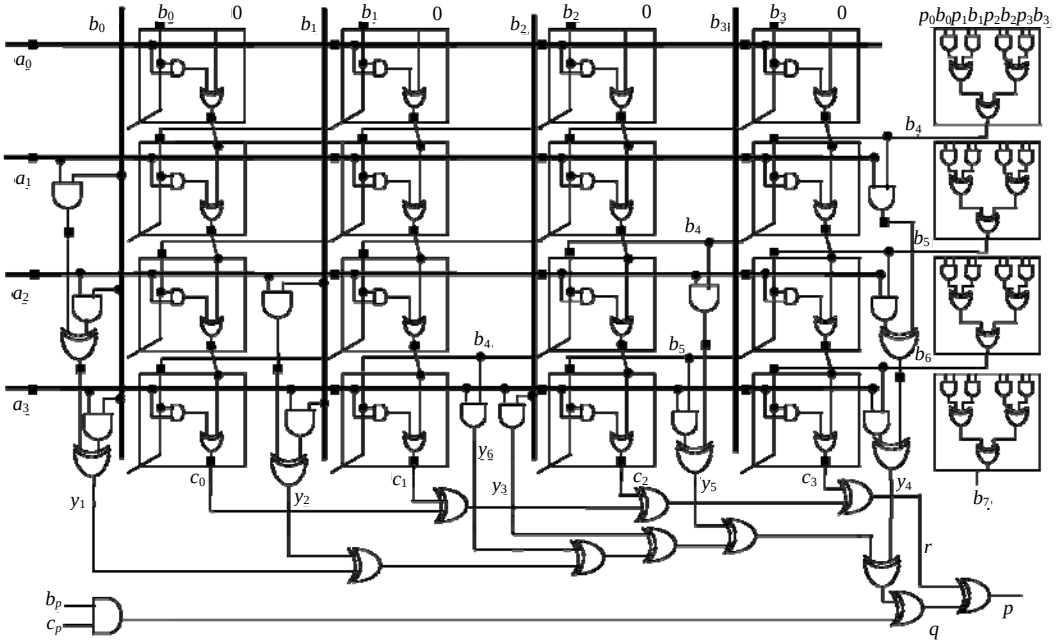

Fig. 4. A parity checking circuit for the bit-parallel systolic multiplication over GF(2^4) using dual base.

Now, rearranging, we see that q and r are same:

$$q=b_0a_0\oplus b_1a_1\oplus b_2a_2\oplus b_3a_3\oplus b_1a_0\oplus b_2a_1\oplus b_3a_2\oplus b_4a_3\oplus b_2a_0 \oplus b_3a_1\oplus b_4a_2\oplus b_5a_3\oplus b_3a_0\oplus b_4a_1\oplus b_5a_2\oplus b_6a_3.$$

A parity checking circuit is presented in the figure which is correctly functioning for the Bit-parallel systolic multiplication over GF(2^4) using dual base. If the circuit operation is correct then q and r will agree and $p=r\oplus q=0$. If any cell in the circuit is faulty, it will change the output lines and that fault reflects in the r line, as q remains unaltered, so p=1 and the fault is detected. And if there is any failure in the y_i line it can also be detected by p=1. Actually few of the y_i terms cancel the output parity checking operation because they appear an even number of times in the coefficient of the output and are cancelled out in the parity-checking operation. It can be improved further as the y_i terms are the sum of the results of some of the individual cells. So, if it is possible to temporarily disconnect those cells and connect with some lines to produce the desired feedback lines, the extra gates will not be required for the check line q. Then the circuit complexity will be reduced and less time will be required.

Delay: As the architecture is pipelined, so the path delays of each stage is same, except the last stage. The last has the maximum path delay. This can be calculated as for m-bit architecture. So,

$$T_d=2mT_{\text{XOR}}+T_{\text{AND}}.$$

In our example in Fig. 1, we calculate the path delay as $T_d=8T_{\text{XOR}}+T_{\text{AND}}$.

5. Simulation Result

We have modeled our proposed architecture in VHDL. The design was simulated in "Model Sim XE III 6.3c" and checked the functionality of the multiplier for different values of m. The physical synthesis and place and route are done using Magma design Automation EDA tools based on Austria Microsystems 0.35 micron technology. The post CTS-post detailed route layout of design for GF(2^4) is

shown in Fig. 5.

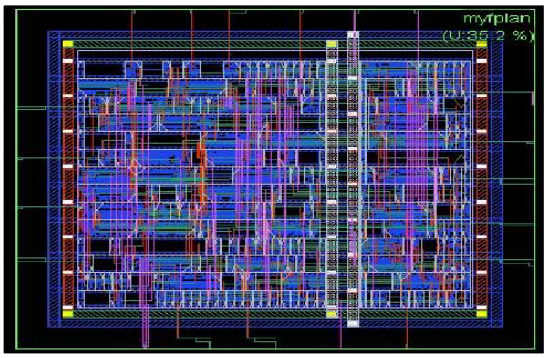

Fig. 5. Layout of bit-parallel dual basis systolic multiplier for GF(2^5) with error checking circuit.

6. Conclusions

The paper presented a fast dual-basis error tolerant bit-parallel systolic multiplier architecture over GF(2^m), which can be pipelined and which requires less hardware compared with the multiplier architecture proposed earlier. Our proposed multiplier can also operate over both the dual-base and polynomial base. The proposed multiplier provides shorter longest delay path compared with earlier architecture. A simple and efficient error detection procedure using parity checking has been incorporated with some additional AND-XOR gates.

Acknowledgment

This work has been supported in part by a Royal Society (UK) International Incoming Fellowship awarded to Dr. Hafizur Rahaman.

References

[1] T. A. Gulliver, M. Serra, and V. K. Bhargava, "The generation of primitive polynomials in GF(2^m) with independent roots and their application for power residue codes, VLSI testing and finite field multipliers using normal bases," *Intl. J. Electronics*, vol. 71, no. 4, pp. 559-576, 1991.

[2] R. E. Blahut, *Fast Algorithms for Digital Signal Processing*, Reading, Mass: Addison Wesley, 1985.

[3] E. R. Berlekamp, "Bit-serial Reed-Solomon encoders," *IEEE Trans. Inf. Theory*, 1982, vol. 28, no. 6, pp. 869-874, 1982.

[4] I. S. Hsu, T. K. Truong, L. J. Deutsch, and I. S. Reed, "A comparison of VLSI architectures of finite field multipliers using dual, normal or standard bases," *IEEE Trans. on Computers*, vol. 37, no. 6, pp. 735-737, 1988.

[5] C. H. Kim, C. P. Hong, and S. Kwon, "A digit-serial multiplier for finite field GF(2^m)," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 4, pp. 467-483, 2005.

[6] M. K. Hasan and V. K. Bhargava, "Division and bit-serial multiplication over GF(q^m) ," *IEE Proc. E*, vol. 139, no. 3, pp. 230-236, 1992.

[7] K. W. Kim, K. J. Lee, and K. Y. Yoo, "A new digit-serial systolic multiplier for finite fields GF(2^m)," in *Proc. of 2001 Intl. Conf. on Info-Tech and Info-Net*, Beijing, 2001, pp. 128-133.

[8] C. S. Yeh, I. S. Reed, and T. K. Truong, "Systolic multi-pliers for finite fields GF(2^m)," *IEEE Trans. on Computers*, vol. 33, no. 4, pp. 357-360, 1984.

[9] L. S. Reed and X. Chen, *Error-Control Coding for Data Networks*, Norwell, USA: Kluwer Academic, 1999.

[10] S. T. J. Fenn, M. Benaissa, and D. Taylor: "Dual basis systolic multipliers for GF(2^m)," *IEE Comp. Digit. Tech.*, vol. 144, no. 1, pp. 43-46, 1997.

[11] C. K. Koc and B. Sunar, "Mastrovito multiplier for all trinomial," *IEEE Trans. on Computers*, vol. 48, no. 5, pp. 522-527, 1999.

[12] E. D. Mastrovito, "VLSI Architectures for computation in Galois fields," Ph.D. disseration, Linkoping Univ., Sweden, 1991.

[13] S. T. J. Fenn, M. Benaissa, and D. Taylor, "GF(2^m) multiplication and division over the dual basis," *IEEE Trans. on Computers*, vol. 45, no. 3, pp. 319-327, 1996.

[14] C. L. Wang and J. L. Lin, "Systolic array implementation of multipliers for GF(2^m)," *IEEE TCAS*, vol. 38, no. 7, pp. 796-800, 1991.

[15] S. T. J. Fenn, M. Benaissa, and D. Taylor, "Bit-serial dual basis systolic multipliers for GF(2^m)," *IEEE International Symposium on Circuits and Systems*, Seattle, Washington, USA, 1995, pp. 2000-2003.

[16] H. Rahaman, J. Mathew, D. K. Pradhan, and A. M. Jabir, "C-testable bit parallel multipliers over GF(2^m)," *ACM Trans. on Design Automation of Electronic Systems (TODAES)*, vol. 13, no. 1, pp. 1-18, 2008.

[17] R. Furness, M. Benaissa, and S. T. J Fenn, "Generalized triangular basis multipliers for the design of reed-solomon codes," in *Proc. of IEEE Workshop on Signal Processing Systems*, Leicester, UK, 1997, pp. 202-211.

[18] S. T. J. Fenn, M. Benaissa, and D. Taylor, "Division in GF(2^m)," *Electron. Letter*, vol. 28, pp. 2259-2261, Nov. 1993.

[19] S. Kumar, T. Wollinger and C. Paar, "Optimum digit serial GF(2^m) multipliers for curve-based cryptography," *IEEE Trans. on Computers*, vol. 55, no. 10, pp. 1306-1311, 2006.

Ashutosh Kumar Singh received the Ph.D. degree in electronics engineering from Banaras Hindu University, India, in 2000. Currently he is a faculty member with the Department of ECEC, School of Engineering and Science, Curtin University of Technology, Miri, Malaysia. He has published more than 50 research papers in different conferences and journals in these areas. He is a co-author of two books: *Digital Systems Fundamentals* and *Computer System Organization and Architecture* (Prentice Hall). His research interests include verification, synthesis, design, and testing of digital circuits.

Asish Bera received B.E. degree in computer science and engineering from the University of Burdwan, India, in 2007 and is currently pursuing M.S. degree in VLSI design with School of VLSI Technology, Bengal Engineering and Science University, Shibpur, India. His research interests include VLSI Architecture for finite field Arithmetic.

Hafizur Rahaman received his Ph.D. degree in computer science and engineering in 2003 from Jadavpur University, Calcutta, India. He is currently a professor of information technology in Bengal Engineering and Science University, Shibpur, India. His research interest includes logic synthesis and testing of VLSI circuits, fault-tolerant computing, design and testing of Galois field arithmetic circuits. He served in the organizing and programme committee of the International Conference on VLSI Design in 2000 nd 2005, and 2005 Asian Test Symposium (ATS), 2007 IEEE VLSI Design and Test Workshop (VDAT). He is a Member of the IEEE, the IEEE Computer Society, and ACM Sigda.

Jimson Mathew received the Ph.D. degree in computer science in 2008 from University of Bristol, UK Since 2005, he has been with Department of Computer Science, University of Bristol, UK. His research interests primarily focuses on sigma delta converters, fault-tolerant computing, low power design and testing, and Galois field based arithmetic.

Dhiraj K. Pradhan is a currently professor in computer science at the University of Bristol (U.K.). Prior to this, he held a professorship at the University of Massachusetts, Amherst, where he also served as Coordinator of Computer Engineering. He has also worked at the University of California, Berkeley, Oakland University (Michigan), and the University of Regina, in Saskatchewan, Canada. Prof. Pradhan has contributed to very large scale integrated computer-aided design and test, as well as to fault-tolerant computing, computer architecture and parallel processing research, with major publications in journals and conferences, spanning more than 30 years. During this long career, he has been well-funded by various agencies in Canada, USA and UK.