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

A. K. Singh
CS Dept., School of Engineering
Curtin University of Technology, Malaysia

Asish Bera
School of VLSI Technology
Bengal Engg. & Sc. University, Shibpur, India

H. Rahaman, J.Mathew and D.K.Pradhan
Computer Science Dept.
University of Bristol, UK
({hafizur, jimson, pradhan} @cs.bris.ac.uk)

***Abstract*— This paper presents an error tolerant hardware efficient VLSI architecture for bit parallel systolic multiplication over dual base, which can be pipelined. This error tolerant architecture is well suited to VLSI implementation because of its regularity, modular structure, and unidirectional data flow. 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 0.35um CMOS technology. This architecture can also operate over both the dual-base and polynomial base.**

***Keywords*- Finite Field, RS codes, bit parallel, systolic, error correction, VLSI Testing.**

I. INTRODUCTION

Finite field also known as Galois Field arithmetic operations over $GF(2^m)$ find increasing applications in public-key cryptography, error detecting and correcting code[9], VLSI testing[10], digital signal processing[11]. 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 [18-19]. 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 architecture for polynomial basis (PB) multiplier is systolic array architecture. In this type of architecture, a basic cell is repeated in an array and signals flow unilaterally between neighbours. PB systolic array multipliers in $GF(2^m)$ can be classified into four categories, namely bit serial, bit-parallel, hybrid and digit-serial. 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 [12]. This multiplier requires less hardwire. PB bit-serial and bit-parallel systolic multipliers were presented in [8, 13]. A bit-serial dual basis systolic multiplier over $GF(2^m)$ was presented in [3], which requires higher hardware compared to that needed for multiplier proposed in [6] and does not support pipelining. To support pipelining, a modified version which requires less hardware is presented in [14]. 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 [7] 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 [2]. A testable polynomial basis bit parallel multiplier circuits over $GF(2^m)$ was presented in [21]. Although bit-serial dual basis multipliers have been widely employed in applications such as RS encoders [3], it has been proven in [19] that it is advantageous of employing bit-parallel dual basis multipliers, particularly in more complex circuits such as RS decoders and syndrome calculators. Bit-parallel dual basis multipliers therefore allow for 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.

This work was supported in part by Royal Society (UK) Grant.

II PRELIMINARIES

a) 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 $p_i \in GF(2)$ [9]. 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 $A \in GF(2^m)$ 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 $a_i \in GF(2)$. Given A, $B \in GF(2^m)$, 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.

b) Dual Basis Multiplication

Let F_p^m denote the set of all linear function $f: GF(p^m) \rightarrow GF(p)$. A well known linear function is the trace function which is frequently used to produce the finite field multipliers. Rather than trace function there are a number of other linear function. We follow the definition of the duality of two bases [16-17] as given below.

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

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

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 [16] but has subsequently

978-1-4244-2587-7/09/$25.00 ©2009 IEEE

been used to describe finite-field multiplication [15]. Furthermore, as observed in [1], the following represents a generalized and alternative representation of Berlekamp bit-serial multiplier.

Theorem 1 [14]: Let *a, b,* c $\in$ *GF(p^m)* such that $c = ab$. Further, let α be a root of the defining irreducible polynomial for the field, let $\beta \in$ *GF(2^m)*, $f \in F_2^m$ *and r*epresent c over the polynomial basis by $a = \sum_{i=0}^{m-1} a_i \alpha^i$,

Then the following relation holds.

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

We have modified eqn. (1) as follows.

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

Where $b_k = f(b\beta\alpha^k)$ ($k = 0, 1, \ldots 2m - 2$) 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 *eqn.* 1 are the dual-basis coefficients of c and b, respectively. Thus to make use of eqn. 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 $b_m = f(b\beta\alpha^m) = f(b(\beta\sum_{j=0}^{m-1} p_j\alpha^j)) = \sum_{j=0}^{m-1} p_j f(b\beta\alpha^j) = \sum_{j=0}^{m-1} p_j b_j$

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

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

Then in general $b_{m+k} = \sum_{j=0}^{m-1} p_j b_{j+k}$ (3)

where b_k *(k* = 0, 1, ..., *m* - 1) are the dual basis coefficients of b and α is root of p(x). Having generated these values of b_k from eqn. 2 one need to carry out the matrix multiplication given in eqn. 1. Now consider the implementation of this multiplication algorithm in the design of a bit-parallel systolic multiplier.

III BIT PARALLEL DUAL BASIS MULTIPLIER

a) ***Proposed Architecture***

Let *a, b,* c $\in$ *GF(2^m)* such that $c = a\ b$ *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 $a = \sum_{i=0}^{m-1} a_i\alpha^i$, . We can derive followings from eqn. (2). $c_0 = b_0a_0 + b_1a_1 + \ldots + b_{m-1}a_{m-1}$; $c_1 = b_1a_0 + b_2a_1 + \ldots + b_ma_{m-1}$; ... ...; $c_{m-1} = b_{m-1}a_0 + b_ma_1 + \ldots + b_{2m-2}a_{m-1}$

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

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

all that changes in these functions is the value of *k.*

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 eqn. (3) and also introduce a cell-2 for generating eqn. (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 eqn. *(4),* we can write as follows: $h(b, a) = b_ka_0 + b_{k+1}a_1 + b_{k+2}a_2 + b_{k+3}a_3$ ... (5)

This equation can be implemented by the circuit as shown in Fig. 2. From $p(x) = x^4 + x + 1$ and eqns. (3) and (4), we can derive the values of b_4, b_5, b_6 as follows: $b_4 = b_1 + b_0$; $b_5 = b_2 + b_1$; $b_6 = b_3 + b_2$

The eqn. (2) for this example is given below.

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

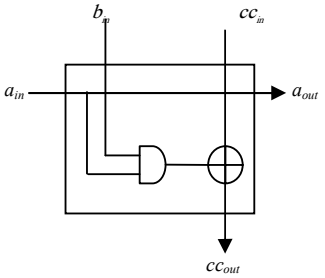

Fig. 1 Generation of Partial Products of eqn. 1

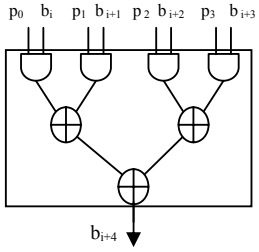

Fig.2: Generation of the sum of partial products of eqn. (2)

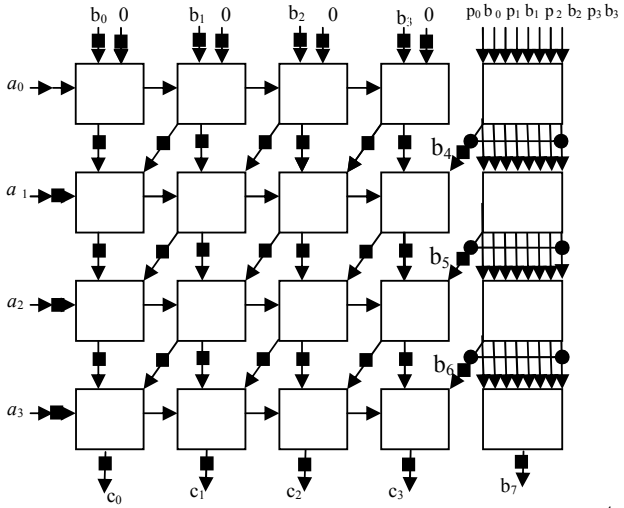

Fig.3: Arrangement of systolic cells for bit-parallel multiplier for GF(2^4).

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, b_6, is generated by the block diagram of Fig.2. In general Fig.2, represents the equation (2) [$b_{m+k} = \sum_{j=0}^{m-1} p_j b_{j+k}$, where $k = 0, 1, \ldots, m - 2$]. The partial sum in the matrix multiplication in eqn. (1) is generated by the block diagram of Fig. 1.

In BP Systolic dual basis multiplier design of [14], there exist two datapath, one is horizontal and the other is vertical. The vertical datapath generates partial sum in matrix multiplication of eqn.(1). The horizontal data path generates partial sum of eqn.(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 $O(\log_2 m)$. For this purpose, we introduce a new cell [Fig. 2] to generate the eqn. (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.

b) Hardware and Delay Analysis

We compare our proposed architecture with the bit parallel architecture described in [16]. 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. First cell consists of one AND gate and one EXOR gate. 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.

Table 1: Comparison between two bit-parallel systolic multipliers

Properties		Reference [19]	Presented here
Number of cells		m^2	cell 1:m^2 & cell 2:m
Circuit complexity	No of 2 input AND gate	$2\ m^2$	$2m^2$
	No of 2 input XOR gate	$2m^2$	$2m^2$ - m
Largest delay path		(2m-1) [D_A + D_X]	m D_A +($\log_2^m$+m-1) D_X

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 the eqn. (6). Longest Delay = $[\{D_A(m+1-1)+D_X(\log_2^m+m-1)]$

$$= \{m\ D_A +(\log_2^m+m-1)\ D_X\} \quad \ldots(6)$$

BP multiplier of [16] has a longest delay path of (2m-1) [D_A+ D_X], whereas the proposed multiplier has a longest delay path of {m D_A+ ($\log_2^m$+m-1) D_X}. Hence, the proposed dual basis BP multiplier is hardware efficient and faster.

Table 2: Hardware requirements and delays of dual basis Bit parallel multiplier (DPM) presented in [16] and the proposed multiplier (DPM)

	DPM [16]			DPM [PROPOSED]		
m	AND	XOR	Delay	AND	XOR	delay
2	8	8	3[D_A+D_X]	8	6	2D_A+2 D_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

From the table we can conclude that in this architecture, the number of AND gates are same compared to previous architecture [19], but for m-bit dual basis systolic multiplier m no. 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.

In Table 2, the hardware complexity and delays of the DPM [19] and the our proposed DPM architecture are given for $GF(2^m)$ for $(m = 2, 3, \ldots, 10)$. From Table 1, it can be seen that for every case, the hardware complexity and delays of our proposed DPM architecture are less compared to those of the DPM architecture [19].

IV. 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,

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

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

Now, from eqn. (2) of the previous architecture, we get

$c_0 = b_0a_0 \oplus b_1a_1 \oplus b_2a_2 \oplus b_3a_3$; $c_1 = b_1a_0 \oplus b_2a_1 \oplus b_3a_2 \oplus b_4a_3$; $c_2 = b_2a_0 \oplus b_3a_1 \oplus b_4a_2 \oplus b_5a_3$; $c_3 = b_3a_0 \oplus b_4a_1 \oplus b_5a_2 \oplus b_6a_3$

Now, we denote the modulo2 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 the 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 bp.cp and the y_i lines.

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

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

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

$q = b_0a_0 \oplus b_1a_1 \oplus b_2a_2 \oplus b_3a_3 \oplus b_1a_0 \oplus b_2a_1 \oplus b_3a_2 \oplus b_4a \oplus b_2a_0 \oplus b_3a_1 \oplus b_4a_2 \oplus b_5a_3 \oplus b_3a_0 \oplus b_4a_1 \oplus b_5a_2 \oplus b_6a_3$

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, that 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 that can also be detected by p=1. Actually few of the y_i terms cancel the output parity checking operation as because they appear an even number of times in the coefficient of the output and 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 then 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, $T_d = 2mT_{XOR} + T_{AND}$

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

a) *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.

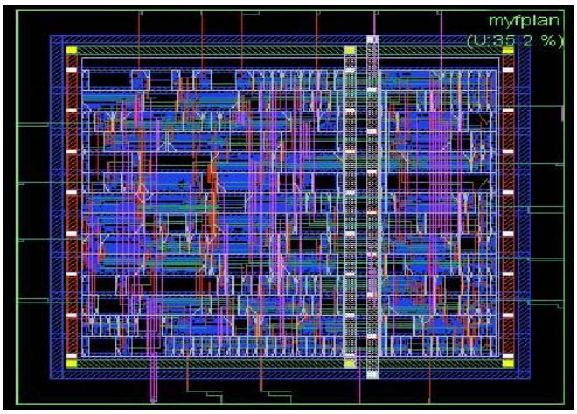

Fig.5: Layout of Bit-parallel Dual Basis systolic Multiplier for GF (2^5) with Error Checking Circuit

V. 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 to that required in 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 to that provided by the architecture presented earlier. A simple and efficient error detection procedure using parity checking has been incorporated with some additional AND- XOR gates.

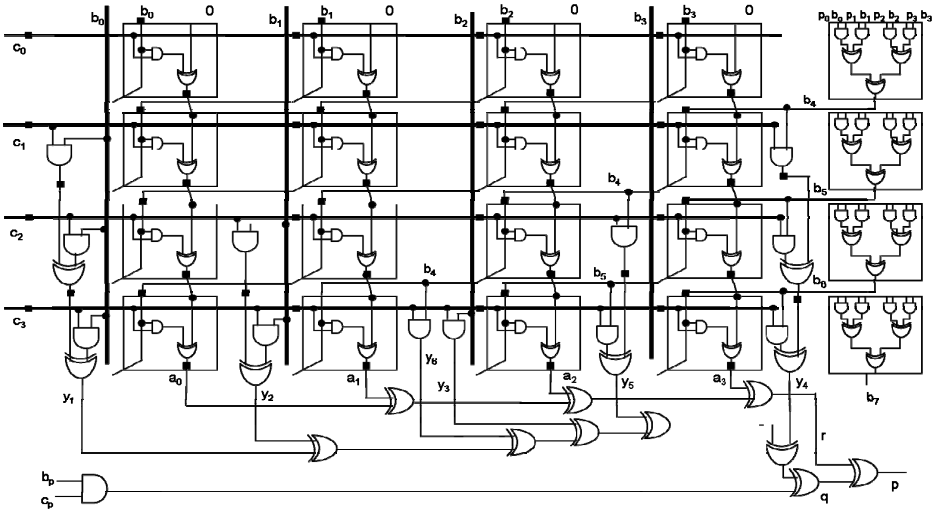

Fig 4: A parity checking circuit for the bit-parallel systolic multiplication over GF (2^4) using dual base.

REFERENCE

1. S.Kumar,T.Wollinger, and C.Paar, "Optimum Digit Serial GF(2^m) Multipliers for Curve-based Cryptography",*TC*, vol.55(10), pp.1306-1311, 2006.
2. S.T.J. Fenn, M. Benaissa, and D. Taylor, "Bit-Serial Dual Basis Systolic Multipliers for GF(2^m)", *ISCAS 1995*, vol.3, pp.2000-2003.
3. C. K. Koc and B. Sunar, "Mastrovito Multiplier for all Trinomial", *IEEE Transactions on Computers*, vol. 48, No.5, pp.522-527, May 1999.
4. M. K. Hasan and V. K. Bhargava, "Division and bit-serial multiplication over *GF(q^m)*", *IEE Proc*. E, May 1992,139(3), pp.230-236.
5. R. Furness, M. Benaissa and S.T.J Fenn, "Generalized Triangular Basis Multipliers for the Design of Reed-solomon Codes", *IEEE Workshop on Signal Processing Systems*, 1997, pp.202-211.
6. E. D. Mastrovito, *"VLSI Architectures for Computation in Galois Fields*", PhD thesis, Linkoping Univ, Sweden, 1991.
7. C. L. Wang and J. L. LIN, "Systolic Array Implementation of Multipliers for *GF(2^m)"*, *IEEE TCAS*, 1991, Vol.38(7), pp 796-800.
8. L.S. Reed and X.Chen, *Error-Control Coding for Data Networks*, Kluwer Academic, 1999.
9. T.A. Gulliver, M. Serra, and V.K. Bhargava, "The Generation of Primitive Polynomials in *GF(2m)* with Independent Roots and Their Application for Power Residue Codes, VLSI Testing and Finite Field Multipliers Using Normal Bases," *Int'l J. Electronics*, vol. 71, no. 4, pp. 559-576, 1991.
10. R.E. Blahut, *Fast Algorithms for Digital Serial Processing.* Addison Wesley, 1985.
11. Berlekamp, E.R.: 'Bit-serial Reed-Solomon encoders', *IEEE Trans. Inf. Theory,* 1982, 28, (6), pp. 869-874
12. Yeh, C.S., Reed, I.S., and Truong, T.K.: 'Systolic multi-pliers for finite fields GF (2^m)', *IEEE TC.,* 1984, vol.33(4), pp. 357-360
13. S.T. J. Fenn, M. Benaissa, and D.Taylor: "Dual basis systolic multipliers for *GF(2^m)*", *IEE Comput. Digital. Tech. Vol. 144, No.1, January 1997.*
14. Fenn, S.T.J., Benaissa, M., and Taylor, D.: 'GF (2m) multiplication and division over the dual basis', *IEEE TC.,* 1996, 45, (3), pp. 319-327.
15. Fenn, S.T.J., Benaissa, M., and Taylor, D.: 'Division in GF (2^m)', *Electron. Letter*, 1993, 28, pp. 2259-2261.
16. Wang, C.L., and Lin, J.L.: 'Systolic array implementation of multiplier for finite fields GF (2^m)', *IEEE TCAS*-38(7), pp. 796-800, 1991.
17. Fenn, S.T.J., Benaissa, M., and Taylor, D.: 'GF (2^m) multiplication and division over the dual basis', *IEEE TC*, 1996, 45**,** (3), pp. 319-327.
18. 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 TC,*1988, Vol.37(6), pp.735-737.
19. C. H. Kim, C.P. Hong and S. Kwon, "A Digit-Serial Multiplier for Finite Field GF(2^m)", *IEEE TVLSI,* vol.13(4), pp.467-483, Apr. 2005.
20. K. W. Kim, K. J. Lee and K. Y. Yoo, "A new digit-serial systolic multiplier for finite fields GF(2^m)", ICII 2001, Beijing, vo.l.5, pp.128-133, Nov.2001.
21. H. Rahaman, J. Mathew, and D. K. Pradhan, "C-testable bit Parallel Multipliers over *GF(2^m)"*, *VLSI Design 2007*, India.