POLYNOMIAL TRANSFORMER

Takakazu Kurokawa and Hideo Aiso

Dept. of Electrical Engineering Keio University
3-14-1 Hiyoshi, Yokohama 223, Japan

Abstract

Any relations among finite fields can be transformed to a unique polynomial of one variable using Galois Fields. In this paper, we explain the design for a "Polynomial Transformer" which executes the transformation.

Polynomial Transformer consists of very simple and iterative logic, and it is very suitable for parallel and pipelined VLSI algorithm. Moreover, three dimensional construction of a Polynomial Transformer is possible. Thus, it serves as an example of a typical three dimensional VLSI.

Its application can be found in Polynomial Transformation, disturbance of data and so on.

1. Introduction

We have many application problems in manipulating n functions of m variables such as is done with decoders, multiplexors, code transformers and so on.

As an example, the truth table for a 1-bit full adder is shown in Table 1-1. Using Boolean algebra, we can express the relations by two output functions in the minterm form as follows:

$$\begin{cases} S = \bar{A}\bar{B}C_i+\bar{A}B\bar{C}_i+A\bar{B}\bar{C}_i+ABC_i \\ C_o = \bar{A}BC_i+A\bar{B}C_i+AB\bar{C}_i+ABC_i \end{cases} \quad (1\text{-}1)$$

Table 1-1 Truth table for a 1-bit Full Adder

A	B	C_i	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

But in adopting the notion of Galois fields [1][2], all the relations can be expressed as the following unique polynomial of one variable x [2].

$$f(x)=\alpha^{43}x+\alpha^{58}x^2+\alpha^{39}x^3+\alpha^{46}x^4+\alpha^{30}x^5+\alpha^{57}x^6 \quad (1\text{-}2)$$

where $f(x)\in GF(2^2)$, $x\in GF(2^3)$, α is the primitive element of $GF(2^6)$ and its indexing polynomial is x^6+x+1.

This method of using Galois fields has the advantage of manipulating some functions as a unique function instead of n Boolean functions. In this paper, we call this transformation from n functions of m variables to a unique polynomial of one variable x "Polynomial Transformation", and also call the object which executes the Polynomial Transformation "Polynomial Transformer".

First, we will show the algorithm of Polynomial Transformation, and then show the construction method of Polynomial Transformer suitable for three dimensional VLSI algorithm which requires the following four items.

1) Repetition of simple cells
2) Regularity of data and control flow
3) Parallel and pipelined processing
4) Few and simple contacts between layers

We will also show some applications.

2. Polynomial Transformation

Let us consider the relationships among m inputs and n outputs. We can correspond each m-input vector $(x_1,x_2,\cdots,x_m)$ to an element in $GF(2^m)$ and each n-output vector $(y_1,y_2,\cdots,y_n)$ to an element in $GF(2^n)$, where $x_i,y_j\in GF(2)(i=1,2,\cdots,m;j=1,2,\cdots,n)$. The polynomial according to the input-output functions is given in the extension field $GF(2^l)$, where $l = LCM(m,n)$, as follows:

<Theorem> [2]
Let a polynomial function be expressed as

$$f(x) = a_0+a_1x+a_2x^2+\cdots+a_rx^r \quad (2\text{-}1)$$

where $r = 2^m - 1$
$x \in GF(2^m)$, $y=f(x)\in GF(2^n)$
$a_i\in GF(2^l)$ for $i=0,1,2,\cdots,r$
$l = LCM(m,n)$

CH2146-9/85/0000/0153$01.00

Then the coefficients $\{a_0, a_1, \cdots, a_r\}$ are calculated as follows:

$$a_0 = f(\alpha^{\infty}) \qquad (2\text{-}2)$$

$$a_i = \sum_{x \in GF(2^1)} x^{r-i} f(x) \quad \text{for } i=1,2,\cdots,r \qquad (2\text{-}3)$$

*

This theorem is based on the indeterminate coefficient method. The relation among $x \in GF(2^m)$, $y \in GF(2^n)$ and their extension field $GF(2^l)$ is shown in Fig.2-1.

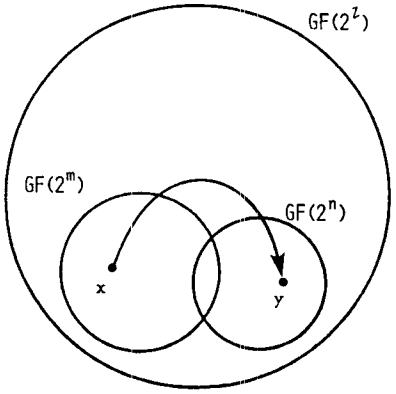

Fig. 2-1 The relation among $x \in GF(2^m)$, $y \in GF(2^n)$ and $GF(2^l)$

<Example>

We will find a polynomial in Galois field which expresses the relationships of Table 1-1. The minimal extension field of $GF(2^3)$ and $GF(2^2)$ is $GF(2^6)$ ($l = 6 = LCM(3,2)$).

Let us choose x^6+x+1 as the primitive polynomial in $GF(2^6)$, and α be the primitive element of the polynomial x^6+x+1, that is, $\alpha^6+\alpha+1=0$. Table 2-1 shows the cyclic representation of $GF(2^6)$.

Table 2-1 The cyclic representation of $GF(2^6)$ ($\alpha^6 = 1 + \alpha$)

e	α^0	α^1	α^2	α^3	α^4	α^5	e	α^0	α^1	α^2	α^3	α^4	α^5
∞	0	0	0	0	0	0	31	1	0	1	0	0	1
0	1	0	0	0	0	0	32	1	0	0	1	0	0
1	0	1	0	0	0	0	33	0	1	0	0	1	0
2	0	0	1	0	0	0	34	0	0	1	0	0	1
3	0	0	0	1	0	0	35	1	1	0	1	0	0
4	0	0	0	0	1	0	36	0	1	1	0	1	0
5	0	0	0	0	0	1	37	0	0	1	1	0	1
6	1	1	0	0	0	0	38	1	1	0	1	1	0
7	0	1	1	0	0	0	39	0	1	1	0	1	1
8	0	0	1	1	0	0	40	1	1	1	1	0	1
9	0	0	0	1	1	0	41	1	0	1	1	1	0
10	0	0	0	0	1	1	42	0	1	0	1	1	1
11	1	1	0	0	0	1	43	1	1	1	0	1	1
12	1	0	1	0	0	0	44	1	0	1	1	0	1
13	0	1	0	1	0	0	45	1	0	0	1	1	0
14	0	0	1	0	1	0	46	0	1	0	0	1	1
15	0	0	0	1	0	1	47	1	1	1	0	0	1
16	1	1	0	0	1	0	48	1	0	1	1	0	0
17	0	1	1	0	0	1	49	0	1	0	1	1	0
18	1	1	1	1	0	0	50	0	0	1	0	1	1
19	0	1	1	1	1	0	51	1	1	0	1	0	1
20	0	0	1	1	1	1	52	1	0	1	0	1	0
21	1	1	0	1	1	1	53	0	1	0	1	0	1
22	1	0	1	0	1	1	54	1	1	1	0	1	0
23	1	0	0	1	0	1	55	0	1	1	1	0	1
24	1	0	0	0	1	0	56	1	1	1	1	1	0
25	0	1	0	0	0	1	57	0	1	1	1	1	1
26	1	1	1	0	0	0	58	1	1	1	1	1	1
27	0	1	1	1	0	0	59	1	0	1	1	1	1
28	0	0	1	1	1	0	60	1	0	0	1	1	1
29	0	0	0	1	1	1	61	1	0	0	0	1	1
30	1	1	0	0	1	1	62	1	0	0	0	0	1

This table shows the relationships among exponent (e), which we will call "E-expression", and the polynomial expression of α, which we will call "P-expression". The transformation from E-expression to P-expression is called "E-P transformation" and vice versa.

Let β and γ be the primitive elements of $GF(2^3)$ and $GF(2^2)$ respectively, which are both subfields of $GF(2^6)$. Then we can find relationships among α, β and γ as follows:

$$\beta = \alpha^{\frac{2^6-1}{2^3-1}} = \alpha^9 \quad , \quad \gamma = \alpha^{\frac{2^6-1}{2^2-1}} = \alpha^{21} \qquad (2\text{-}5)$$

And from Table 2-1 the minimal polynomials of β and γ result in the following:

$$\beta^3 = 1 + \beta^2 \, , \quad \gamma^2 = 1 + \gamma \qquad (2\text{-}6)$$

Table 2-2 and Table 2-3 show the cyclic representation of $GF(2^3)$ with β and that of $GF(2^2)$ with γ respectively. Using Table 2-2, Table 2-3 and Eq.(2-5), we can rewrite Table 1-1 to Table 2-4.

Table 2-2 The cyclic representation of $GF(2^3)$ ($\beta^3 = 1 + \beta^2$)

E-expression	P-expression		
	β^0	β^1	β^2
β^{∞}	0	0	0
β^0	1	0	0
β^1	0	1	0
β^2	0	0	1
β^3	1	0	1
β^4	1	1	1
β^5	1	1	0
β^6	0	1	1

Table 2-3 The cyclic representation of $GF(2^2)$ ($\gamma^2 = 1 + \gamma$)

E-expression	P-expression	
	γ^0	γ^1
γ^{∞}	0	0
γ^0	1	0
γ^1	0	1
γ^2	1	1

Table 2-4 Truth table for a 1-bit Full Adder in $GF(2^6)$

INPUTS				OUTPUTS			
α^{∞}	β^{∞}	— 0 0 0		0 0	— γ^{∞}	=	α^{∞}
α^0	β^0	— 1 0 0		1 0	— γ^0	=	α^0
α^9	β^1	— 0 1 0		1 0	— γ^0	=	α^0
α^{18}	β^2	— 0 0 1		1 0	— γ^0	=	α^0
α^{27}	β^3	— 1 0 1		0 1	— γ^1	=	α^{21}
α^{36}	β^4	— 1 1 1		1 1	— γ^2	=	α^{42}
α^{45}	β^5	— 1 1 0		0 1	— γ^1	=	α^{21}
α^{54}	β^6	— 0 1 1		0 1	— γ^1	=	α^{21}

From the Theorem, the following polynomial function should be obtained.

$$f(x)=a_0+a_1x+a_2x^2+a_3x^3+a_4x^4+a_5x^5+a_6x^6+a_7x^7 \quad (2\text{-}7)$$

where $r=2^3-1=7$

$x \in GF(2^3)$, $y=f(x) \in GF(2^2)$

and $a_i \in GF(2^6)$ for $i=0,1,2,\cdots,7$

From the Theorem and Table 2-4, we can calculate the corresponding coefficients as follows:

$$a_0 = f(\alpha^\infty) = \alpha^\infty$$

$$a_1 = \sum_{x \in GF(2^3)} x^6 f(x)$$

$$= (\alpha^\infty)^6 f(\alpha^\infty) + (\alpha^0)^6 f(\alpha^0) + (\alpha^9)^6 f(\alpha^9) + (\alpha^{18})^6 f(\alpha^{18}) + (\alpha^{27})^6 f(\alpha^{27}) + (\alpha^{36})^6 f(\alpha^{36}) + (\alpha^{45})^6 f(\alpha^{45}) + (\alpha^{54})^6 f(\alpha^{54})$$

$$= \alpha^\infty\alpha^\infty + \alpha^0\alpha^0 + \alpha^{54}\alpha^0 + \alpha^{108}\alpha^0 + \alpha^{162}\alpha^{21} + \alpha^{216}\alpha^{42} + \alpha^{270}\alpha^{21} + \alpha^{324}\alpha^{21}$$

$$= \alpha^\infty + \alpha^0 + \alpha^{54} + \alpha^{108} + \alpha^{183} + \alpha^{291} + \alpha^{345}$$

$$= \alpha^\infty + \alpha^0 + \alpha^{54} + \alpha^{45} + \alpha^{57} + \alpha^6 + \alpha^{39} + \alpha^{30}$$

$$= \alpha^{43}$$

The others are calculated in the same way. Consequently, the desired polynomial function is obtained as Eq.(1-2).

3. Polynomial Transformer

In this chapter we will present the construction method of Polynomial Transformer which is suitable for three dimensional VLSI algorithm. The Polynomial Transformer for n functions of m variables receives the function values $\{f(\alpha^\infty), f(\alpha^0), \cdots, f(\alpha^{2^m-2})\}$ and produces the coefficients $\{a_0, a_1, \cdots, a_r\}$, where $r=2^m-1$.

For example, we will design the Polynomial Transformer of four functions of four variables considering the fabrication technology of three dimensional VLSI. In this case, the corresponding polynomial which we wish to obtain is presented as follows:

$$f(x)=a_0+a_1x+a_2x^2+a_3x^3+\cdots+a_{15}x^{15} \quad (3\text{-}1)$$

where $x \in GF(2^4)$, $y=f(x) \in GF(2^4)$

$a_i \in GF(2^4)$ for $i=0,1,2,\cdots,15$

and

$$a_0 = f(\alpha^\infty) \quad (3\text{-}2)$$

$$a_i = \sum_{x \in GF(2^4)} x^{15-i} f(x) \quad \text{for } i=1,2,\cdots,15 \quad (3\text{-}3)$$

So, the Polynomial Transformer may calculate sixteen coefficients according to Eq.(3-2) and Eq.(3-3) when sixteen values $\{f(\alpha^\infty), f(\alpha^0), \cdots, f(\alpha^{14})\}$ are given. And Eq.(3-2) and Eq.(3-3) show that all the coefficients can be calculated simultaneously so that Polynomial Transformation is suitable for parallel processing.

Fig.3-1 shows the three dimensional block diagram of Polynomial Transformer which produces the sixteen coefficients $\{a_0, a_1, \cdots, a_{15}\}$. It consists of three parts; (1) Input Queue, (2) Coefficient Calculator and (3) P-E transformer. We will explain each part in detail.

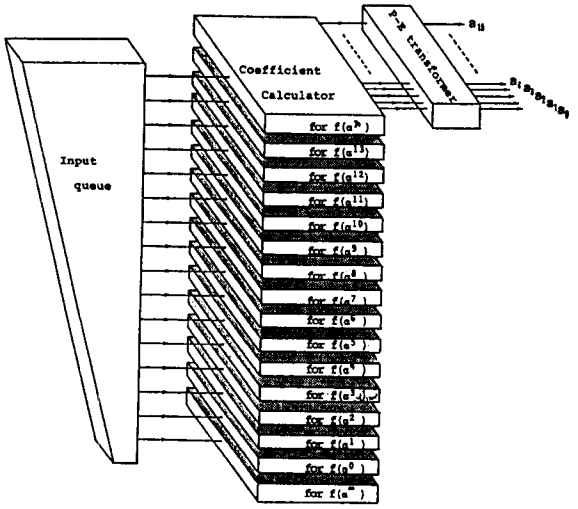

Fig. 3-1 Block diagram of Polynomial Transformer

(1) Input Queue

This queue is provided to pass the input data $\{f(\alpha^i)\}$ to every stage of Coefficient Calculator described below, so that it should keep all the system working synchronously. Input Queue is necessary for the implementation, but it is not essential for the transformation algorithm itself.

(2) Coefficient Calculator

Coefficient Calculator is the substantial body of Polynomial Transformer. The coefficient calculation requires sixteen stages, so that in arranging every stage to a separate layer, we can implement three dimensional Coefficient Calculator as

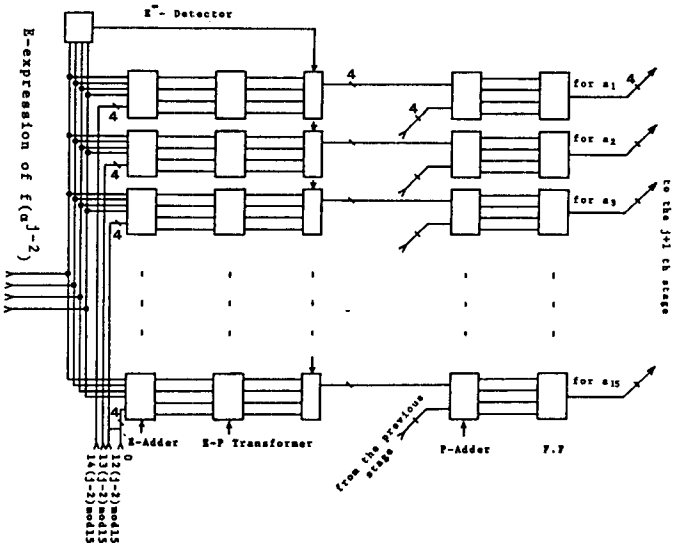

Fig. 3-2 The j-th stage of Coefficient Calculator

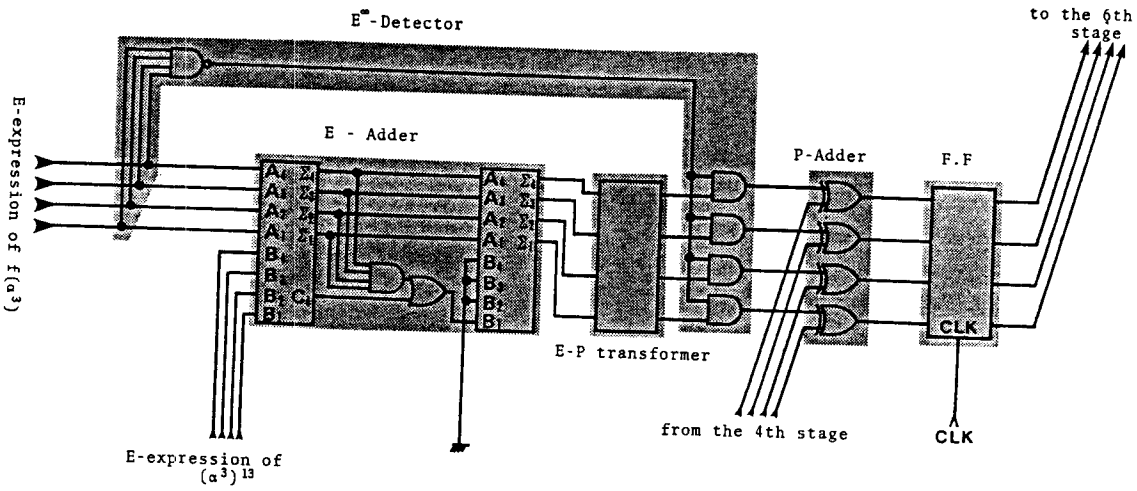

Fig. 3-3 A circuit diagram to calculate the 5th stage for a_2

in Fig.3-1. Fig.3-2 shows the j-th stage of Coefficient Calculator, which calculates the partial sum "+(α j-2)15-i f(α j-2)" for i=1,2,···,15. Fig.3-3 shows the second row (i=2) of the fifth stage (j=5), which calculates the partial sum "+(α3)13 f(α3)".

Coefficient Calculator consists of some modules: 1) E-Adder, 2) E-P Transformer, 3) P-Adder, 4) E∞-Detector and 5) F.F. E-Adder and P-Adder perform the multiplication and addition of elements in GF(2^4) respectively. E-P Transformer is self-explanatory. The circuit diagram of E-P Transformer is illustrated in Fig.3-4. Table 3-1 shows the correspondence between E-expression and P-expression in GF(2^4) where the indexing polynomial is $\alpha^4=1+\alpha$. E∞-Detector detects the α^∞ input which is coded as (1 1 1 1) and truncates the results of E-Adder. F.F is used similarly as Input Queue to pass the partial sum to the next stage synchronously. The coefficient a_0 could be obtained directly as $a_0=f(\alpha^\infty)$ without calculation, but is treated in the same manner as the others $\{a_1, a_2, \cdots, a_{15}\}$ passing through 16 stages.

The first stage consists of only a E-P Transformer of $f(\alpha^\infty)$ for a_0 and a_{15}. The other fifteen stages are all the same as in Fig.3-3.

(3) P-E Transformer

This Transformer transforms the output results of the Coefficient Calculator expressed by P-expression to its E-expression. It is realized similarly as the E-P Transformer described above.

P-E Transformer is not necessary if the coefficients may be out in the form of P-expression.

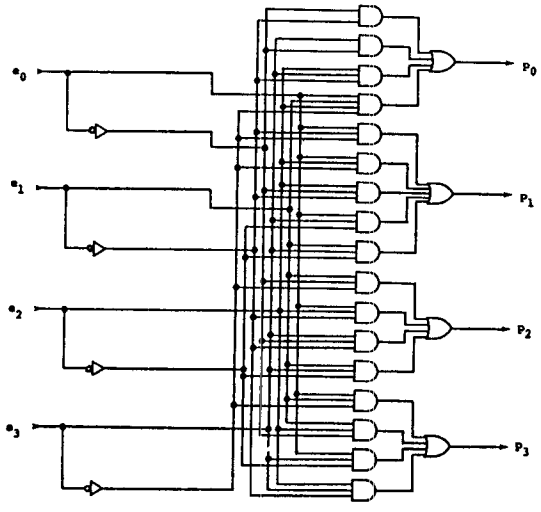

Fig. 3-4 E-P transformer

Table 3-1 The correspondence between E-expression and P-expression in GF(2^4) (indexing polynomial : $\alpha^4 = 1 + \alpha$)

	E-expression	P-expression
α^∞	1 1 1 1	0 0 0 0
α^0	0 0 0 0	1 0 0 0
α^1	0 0 0 1	0 1 0 0
α^2	0 0 1 0	0 0 1 0
α^3	0 0 1 1	0 0 0 1
α^4	0 1 0 0	1 1 0 0
α^5	0 1 0 1	0 1 1 0
α^6	0 1 1 0	0 0 1 1
α^7	0 1 1 1	1 1 0 1
α^8	1 0 0 0	1 0 1 0
α^9	1 0 0 1	0 1 0 1
α^{10}	1 0 1 0	1 1 1 0
α^{11}	1 0 1 1	0 1 1 1
α^{12}	1 1 0 0	1 1 1 1
α^{13}	1 1 0 1	1 0 1 1
α^{14}	1 1 1 0	1 0 0 1

4. Realization of Polynomial Transformer

In this chapter, we discuss the realization of Polynomial Transformer shown in the last chapter.

First we must revise the Coefficient Calculator to make it more compact and faster. In Fig.3-3, E^{∞}-Detector, E-Adder and E-P Transformer can be joined in a combinational network with 4-bit input and 4-bit output because we can know the 4-bit input to E-Adder (B_1,B_2,B_3,B_4) previously. As the result, Polynomial Transformer requires about 18900 logic gates or 87000 transistors in order to be realized. This means that about 1180 logic gates or 5400 transistors should be arranged in every layer.

From this fact and from the simplicity and the iteration of the logic, Polynomial Transformer may be easily realized as a typical three dimensional VLSI. Fig.4-1 shows the layout pattern of the 14th layer of Polynomial Transformer including Coefficient Calculator and Input Queue with standard cell layout on C-MOS technology. The design follows the layout rule shown in [3].

Polynomial Transformer requires about 120 Δ delay to Polynomial Transformation where Δ means the unit gate delay. But with pipelining, it requires only 7 Δ delay to the transformation. This enables Polynomial Transformer to be used as real-time processing.

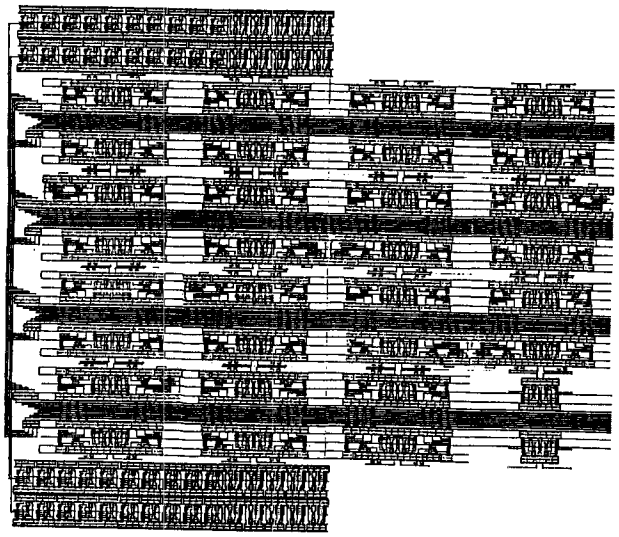

Fig. 4-1 The layout pattern of the 14th layer of Polynomial Transformer

5. Applications

Naturally Polynomial Transformer can be used for Polynomial Transformation with real-time parallel processing. But it also can be used for the disturbance of data which can be used in some fields such as substitution in cipher[4], random number generation[5], hashing[6] and so on. In this chapter we will show the method of data disturbance using Polynomial Transformer constructed in tne former chapter. It can disturb 64-bit data according to the following algorithm.

1) Divide 64-bit data into sixteen blocks of 4-bit data

2) Set up the relations among 4-bit inputs and 4-bit outputs using the results of 1)

3) Execute Polynomial Transformation using Polynomial Transformer

4) Join sixteen coefficients of 4-bit data in 64-bit disturbed data

Fig.5-1 shows an example of data disturbance using Polynomial Transformer. Fig.5-2 and Fig.5-3 shows the usage of Polynomial Transformer to substitution in cipher and random number generation respectively.

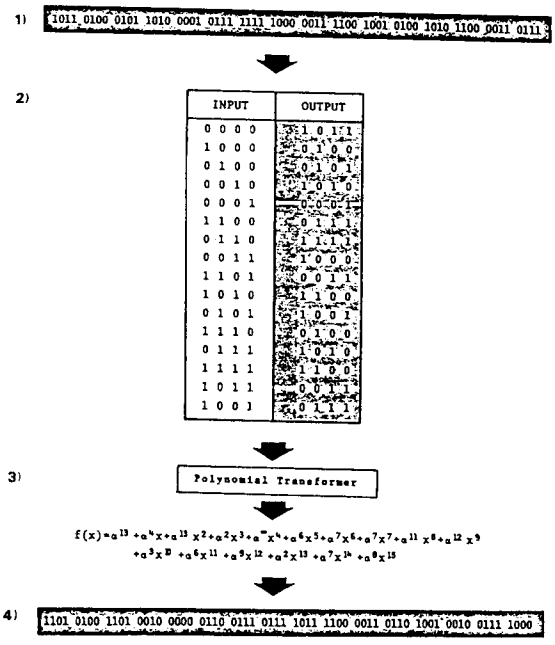

Fig. 5-1 The data disturbance using Polynomial Transformer

6. Conclusion

In this paper, we have proposed Polynomial Transformer and its construction method which is suitable for three dimensional VLSI algorithm. For an example, the layout design of Polynomial Transformer with standard cell layout on C-MOS technology is also shown. Needless to say, in the use of Polynomial Transformation, Polynomial Transformer can be used for real time data disturbance for substitution in cipher, random number generation, hashing and so on.

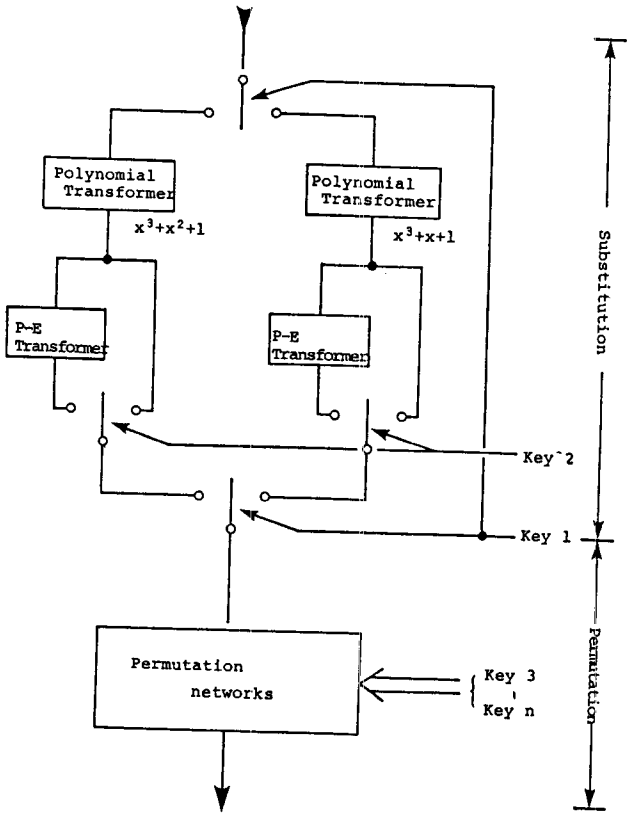

Fig. 5-2 The usage of Polynomial Transformer to substitution in cipher

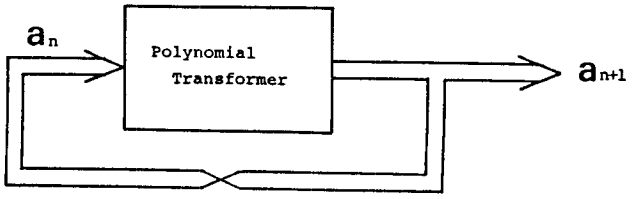

Fig. 5-3 The usage of Polynomial Transformer to random number generation

References

[1] L.E.Dickson : "Linear Groups with an exposition of the Galois Field Theory", Dover Pub., 1958

[2] I.Takahashi : "Combinatorial Theory and its Applications" (in Japanese), Iwanami, 1979

[3] Y.Matsuyama and T.Tomizawa : "Introduction to VLSI design" (in Japanese), Kyoritsu, 1983

[4] Carl H. M. and Stephen M. M. : "Cryptography : A new dimension in computing data security", Wiley-Interscience Pub. 1982

[5] D.E.Knuth : "The Art of Computer Programming, vol.2 : Seminumerical Algorithms" 2nd ed., Addison-Wesley Publishing Co. 1981

[6] D.E.Knuth : "The Art of Computer Programming, vol.3 : Sorting and searching", Addison-Wesley Publishing Co. 1975