A Single Chip VLSI Reed-Solomon Decoder

H. M. Shao, T. K. Truong, I. S. Hsu, and L. J. Deutsch
Communications Systems Research Section

I. S. Reed
University of Southern California

A new VLSI design of a pipeline Reed-Solomon decoder is presented. The transform decoding technique used in a previous design is replaced by a time domain algorithm. A new architecture that implements such an algorithm permits efficient pipeline processing with minimum circuitry. A systolic array is also developed to perform erasure corrections in the new design. A modified form of Euclid's algorithm is implemented by a new architecture that maintains the throughput rate with less circuitry. Such improvements result in both enhanced capability and a significant reduction in silicon area, therefore making it possible to build a pipeline (31,15) RS decoder on a single VLSI chip.

I. Introduction

Recently Brent and Kung (Ref. 1) suggested a systolic array architecture to compute the greatest common divisor (gcd) of two polynomials. Based on this idea a VLSI design of a pipeline Reed-Solomon decoder was developed (Ref. 2). The syndrome computation of this decoder for a 4-bit (15,9) RS code was implemented on a chip (Ref. 3).

In the design of the chip for the above-mentioned decoder, three major problems arose:

(1) While the architecture for syndrome computation took $(N - I)$ cells for an (N, I) RS code, it required N identical cells to implement the inverse transform in the architecture suggested in Ref. 2. As a consequence for a long code such as the (255,223) RS code, the inverse transform circuit would need 255 cells and be quite large.

(2) The basic cell of the systolic array needed to perforr a modified form of Euclid's algorithm occupied considerable silicon area, approximately 60 times the siz of a syndrome computing cell. Since the decodin algorithm in Ref. 2 required $(N - I)$ of such cells, th entire systolic array needed much more silicon are than desired.

(3) Erasure corrections became necessary and were no included in the original design. Hence the decode required several modifications of the original architecture design in Ref. 2.

To reduce the large circuit area required by the invers transform operation it was decided to modify the origina transform decoding algorithm. Also after considering the nee for erasure correction, it was found that the decoding algorithm given in Ref. 4 could accommodate both requirements.

In this algorithm the errata magnitudes are calculated in the time domain and a Chien search is used to find the error locations. The architecture of the new algorithm is designed to operate sequentially in a pipeline, thereby enabling the circuit size to grow with the error correcting capability $(N - I)$ instead of the code length N.

The systolic array designed originally for the modified form of Euclid's algorithm could process polynomials continuously (Ref. 2). However, in real-time RS decoding, there is a need to compute only one syndrome polynomial for each received codeword. If one takes advantage of this by a better utilization of multiplexing, the required pipeline throughput rate can be maintained by the use of fewer basic cells.

In this article, an improved VLSI architecture over that in Ref. 2 is developed utilizing the above observations. A systolic array is also designed for the needed polynomial expansion used in the erasure polynomial computation. These new modifications result in both an enhanced capability and a significant reduction in silicon area without any loss in the pipeline throughput rate.

II. The Decoder Architecture

Let $N = 2^m - 1$ the length of the (N, I) RS code over $GF(2^m)$ with design distance d. Suppose that t errors and s erasures occur, and $s + 2t < d - 1$. The decoding procedure in Ref. 4 is summarized as follows:

Let X_i be an error location or an erasure location and $\Lambda = \{X_i | X_i$ is an erasure location$\}$, $\lambda = \{X_i | X_i$ is an error location$\}$. Let Y_i be the corresponding errata magnitude and $r = (r_0, r_1, \ldots, r_{N-1})$ be the received vector.

Step 1. Compute the syndrome polynomial

$$S(Z) = \sum_{k=1}^{\infty} S_k Z^{-k}$$

where

$$S_k = \sum_{n=0}^{N-1} r_n \alpha^{nk}$$

$$= \sum_{i=1}^{s+t} Y_i X_i^k \qquad (1)$$

for $1 \leqslant k \leqslant d - 1$

Step 2. Compute the erasure locator polynomial

$$\Lambda(Z) = \prod_{X_i \epsilon \Lambda} (Z - X_i) \qquad (2)$$

from Λ.

Step 3. Multiply $S(Z)$ and $\Lambda(Z)$ to obtain the Forney syndrome polynomial

$$T(Z) = S(Z)\Lambda(Z) \qquad (3)$$

Step 4. Compute the errata evaluator polynomial $A(Z)$ and the error locator polynomial $\lambda(Z)$ from $T(Z) = [A(Z)] / \lambda(Z)]$ by the modified Euclid's algorithm.

Step 5. Multiply $\Lambda(Z)$ and $\lambda(Z)$ to get the errata locator polynomial

$$P(Z) = \Lambda(Z)\lambda(Z) \qquad (4)$$

Step 6. Perform Chien search on $\lambda(Z)$ to find the error location set λ.

Step 7. Compute the errata magnitudes

$$Y_k = \frac{A(X_k)}{X_k P'(X_k)}$$

for $1 \leqslant k \leqslant s + t$ by evaluating $A(Z)$ and $P'(Z)$. Use sets λ and Λ to direct the additions of Y_k to the received vector r.

The pipeline architecture of the RS decoder is shown in Fig. 1. The decoder computes the syndrome polynomial $S(Z)$ by the transform circuit given in Ref. 2. The erasure information Λ enters the decoder in the form of a binary sequence.

The systolic array described in the next section expands the factors of

$$\Lambda(Z) = \prod_{X_i \epsilon \Lambda} (Z - X_i)$$

into the polynomial. Polynomial multiplications are performed with a circuit described in Ref. 5. A new architecture is developed which implements the modified Euclid's algorithm by operating on the product of $S(Z)$ and $\Lambda(Z)$. The resulting error locator polynomial $\lambda(Z)$ is then multiplied by $\Lambda(Z)$, thereby obtaining the errata locator polynomial $P(Z)$, The derivative $P'(Z)$ of $P(Z)$ is obtained by dropping the even terms of $P(Z)$. The errata magnitudes Y_k are calculated then by a field inversion and a number of multiplications. Next the error locations are obtained in the form of a binary sequence by the use of another polynomial evaluation circuit which performs the Chien search on $\lambda(Z)$. This sequence of error locations, together with the input erasure location binary sequence, directs the addition of Y_k to the received message.

III. A VLSI Design for Expanding the Erasure Locator Polynomial

It is reasonable to assume that the erasure location information derived from outside the chip, possibly from a convolutional decoder. Let it arrive serially in the form of 1's and 0's. A simple circuit of the form shown in Fig. 2(a) first converts this erasure data into a sequence of α^k's and 0's, where $\alpha^k \in \Lambda$.

Given $\alpha^k \in \Lambda$, the computation of the erasure polynomial demands the expansion of

$$\Lambda(Z) = \prod_{\alpha^k \in \Lambda} (Z - \alpha^k)$$

$$= (Z - \alpha^{k_1})(Z - \alpha^{k_2}) \ldots (Z - \alpha^{k_s}) \qquad (6)$$

Note that for an arbitrary polynomial $Q(Z)$ that

$$Q(Z)(Z - \alpha^k) = ZQ(Z) - \alpha^k Q(Z) \qquad (7)$$

Such an operation involves polynomial shifts, scalar multiplications and additions. Thus the multiplications of $(Z - \alpha^k)$ in Eq. (6) can be implemented by the systolic array given in Fig. 2(b). Since it contains zeros as well as α^k's, the input stream is used to control the updating of the latches in each basic cell. At the end of the arrivals of the erasure locations, the coefficients of $\Lambda(Z)$ are loaded from the latches into registers and shifted out serially.

IV. A New Architecture to Perform the Modified Euclidean Algorithm

A systolic array was designed in Ref. 2 to compute the error locator polynomial by a modified Euclidean algorithm. The array required $2t$ cells, twice the number of correctable errors. It is capable of performing the modified Euclidean algorithm continuously.

In the modified Euclidean algorithm only one syndrome polynomial is computed in the time interval of one code word. As a consequence, for the original architecture in Ref. 2, a pipeline RS decoder is not as efficient as it might be. A substantial portion of the systolic array is always idling. This fact makes possible a more efficient design with fewer cells and no loss in the throughput rate.

For the (N, I) RS code, the length of the syndrome polynomial is $N - I$. The maximum length of the resultant Forney syndrome polynomial is also $N - I$. Imagine now that a single cell is used recursively to perform the successive steps of the modified Euclidean algorithm instead of pipelining data to the next cell. Then it would take $N - I$ recursions to complete the algorithm, where each recursion requires $N - I$ symbol times. Therefore, using a single cell recursively requires only a total of $(N - I)^2$ symbol time to complete the modified form of Euclidean algorithm. Since a syndrome polynomial needs to arrive every N symbol times, only $\lfloor (N - I)^2/N \rfloor$ cells are needed to process successive syndrome polynomials at a full pipeline throughput rate.

Figure 3 shows the new alternate architecture design. The input multiplexer directs the syndrome polynomials to different cells. Each processor cell is almost identical to the cell presented in Ref. 2, except that it is used to process data recursively.

The primary difference in the new cell structure from the architecture of the previous cell (Ref. 2) is presented as follows: Since division is avoided in the modified form of Euclid's algorithm, a scalar factor appears at the output. Although such a scale factor, call it K, is irrelevant to the problem of finding roots of the error locator polynomial $\lambda(Z)$, it must be removed from the errata evaluator polynomial $A(Z)$. In order to effectively utilize the processor cell given in Ref. 2, the factor K which appears at the output of each cell is calculated independently of the cell computation. This is accomplished by using a multiplier, operating recursively, to accumulate the product of all the nonzero leading coefficients of the divisor polynomials. An inverse computation circuit and a multiplier after the demultiplexer is used to remove the unwanted scalar K from $KA(Z)$. This computational process is illustrated in Fig. 3.

The architecture of the new basic cell is given in Fig. 4. Compared with the previous systolic array design (Ref. 2), the present scheme for multiplexing the recursive cell computations significantly reduces the number of cells and as a consequence the number of circuits. Table 1 shows that the cell reduction is greater for high rate codes.

V. A Polynomial Evaluation Pipeline

Polynomials are evaluated not only in the Chien search process, but also when the errata magnitudes are computed. In RS decoding, one needs to evaluate

$$A(Z) = \sum_{i=0}^{S+t-1} A_i Z^i \qquad (8)$$

for $Z = \alpha^k$ and $0 \leqslant k \leqslant N - 1$ given A_i, $0 \leqslant i \leqslant s + t - 1$. Note that Eq. (8) has a form which is identical to the syndrome computation Eq. (1).

However, in Eq. (8) the polynomial is shorter than in Eq. (1). Also since $N > s + t - 1$, Eq. (8) is evaluated over a wider range than Eq. (1) is computed. These two differences make it inefficient to implement Eq. (8) in a manner similar to that used for syndrome computations. A better method is to evaluate $A_i(\alpha^i)^k$ sequentially for each k at cell i. This is illustrated in Fig. 5. The polynomial coefficient A_i is multiplied by α^i at the initialization of cell i. From then on a feedback loop computes the quantities $A_i(\alpha^i)^k$ for $k = 1, 2, 3, \ldots, N - 1$. The summation shown at the bottom of the figure is implemented quite simply since all quantities are binary.

VI. Conclusion

An improved VLSI architecture of a pipeline Reed-Solom decoder is presented herein. Compared with the previo design in Ref. 2, this architecture not only now correc erasures, it is simpler, more regular, smaller in chip area a operates equally as fast. It is estimated that the polynom expansion circuit and the polynomial multiplication circu need approximately the same number of transistors as t syndrome computing pipeline. On the other hand, ea polynomial evaluation circuit takes about half the number transistors. Finally, each cell in the modified form of Euclic algorithm circuit requires approximately the same chip ar as the syndrome circuit.

Based on a previous nMOS chip fabrication of the sy drome pipeline (Ref. 3) and the design of the basic cell of t modified form of Euclid's algorithm (Ref. 6), it is estimat that a (15,9) RS decoder chip would require about 29 tho sand transistors. A (31,15) RS decoder would require abo 88 thousand transistors. Considering the presently existi VLSI technology, a high throughput 5-bit (31,15) RS decod could be implemented readily on a single VLSI chip. Of cour such a chip would have a possible immediate application JTIDS (for Joint Tactical Information Distribution Syste of DoD).

References

1. Brent, R. P., and Kung, H. T., "Systolic VLSI arrays for polynomial GCD computations," Dep. Computer Science, Carnegie-Mellon Univ., Pittsburgh, PA, Rep., 1982.
2. Shao, H. M., Truong, T. K., Deutsch, L. J., Yuen, J. H., and Reed, I. S., "A VLSI Design of a Pipeline Reed-Solomon Decoder," *IEEE Trans. on Computer*, Vol. C-34, No. 5, May 1985, pp. 393-403.
3. Shao, H. M., "A VLSI Syndrome Computing chip for Reed-Solomon Decoding," to be published in *TDA Progress Report 42-85*, Jet Propulsion Laboratory, Pasadena, Calif.
4. Reed, I. S., Truong, T. K. and Miller, R. L., "Decoding of B. C. H. and R. S. codes with errors and erasures using continued fractions," *Electronic Letters*, August 1979, Vol. 15, No. 17, pp. 542-544.
5. Peterson and Weldon, *Error-Correcting Codes*, 2nd Edition, the MIT Press, Cambridge, pp. 172-173, 1972.
6. Hsu, I. S. and Shao, H. M., "A VLSI chip for the implementation of the modified Euclid's Algorithm," to be published in *TDA Progress Report 42-85*, Jet Propulsion Laboratory, Pasadena, Calif.

Table 1. The comparison of the number of cells required in the modified Euclid's algorithm computation

RS Code	Full Systolic Array	Multiplexing on Recursive Calls
(15,9)	6	3
(31,15)	16	9
(255,223)	32	5

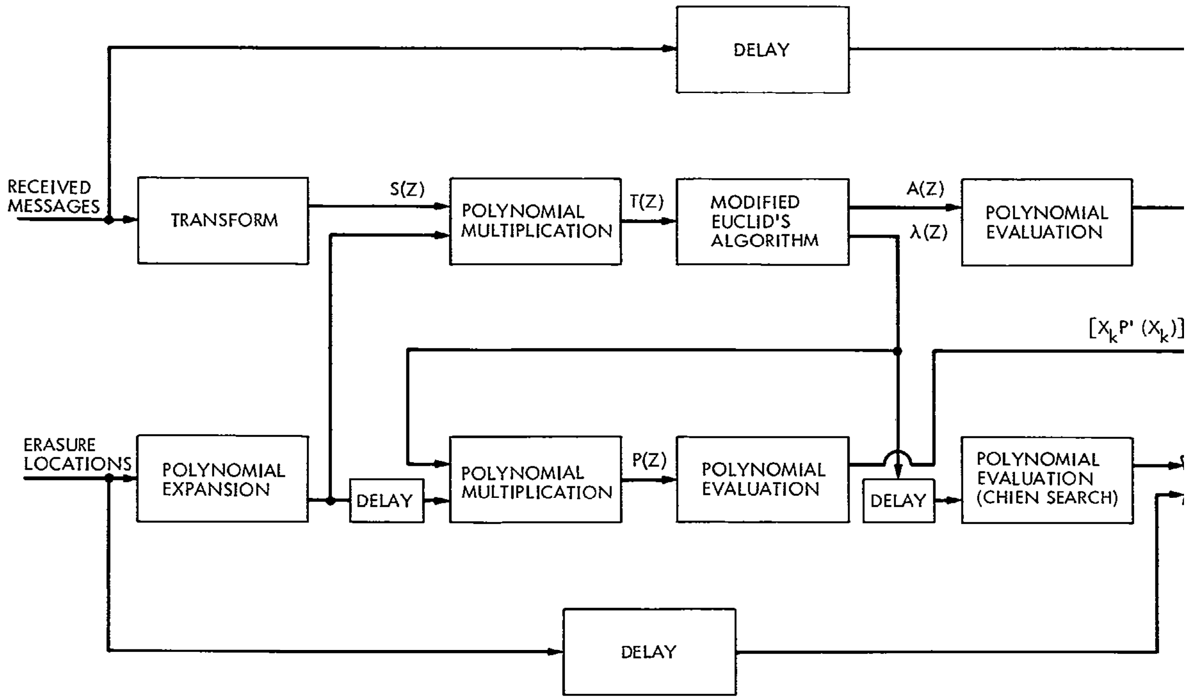

Fig. 1. VLSI architecture of a pipeline Reed-Solomon decoder for both errors and erasures corre

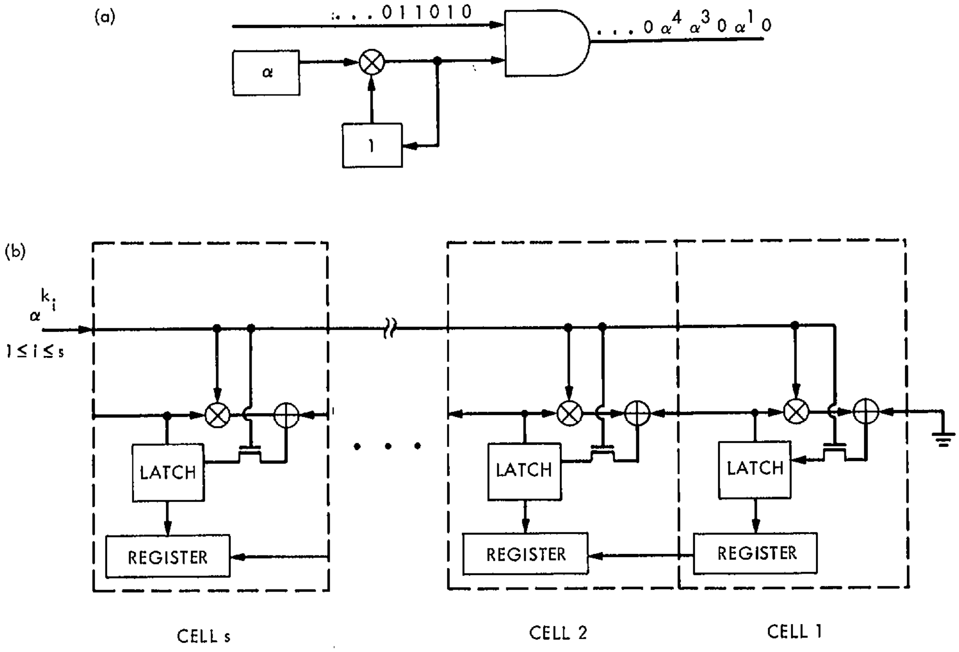

Fig. 2. A systolic array to expand a polynomial

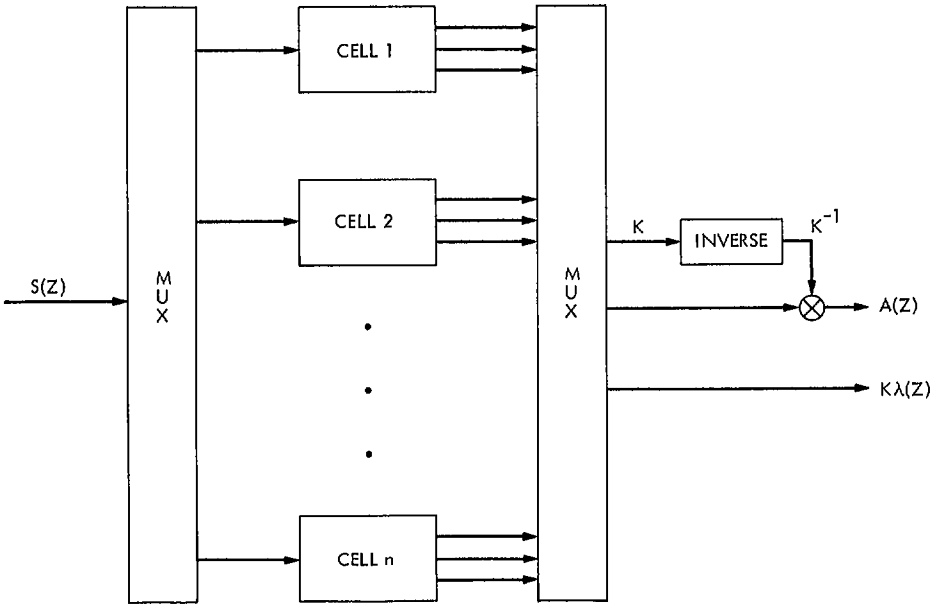

Fig. 3. The new architecture to perform the modified form of Euclid's algorithm where $n = \lfloor (N-I)^2/N \rfloor$

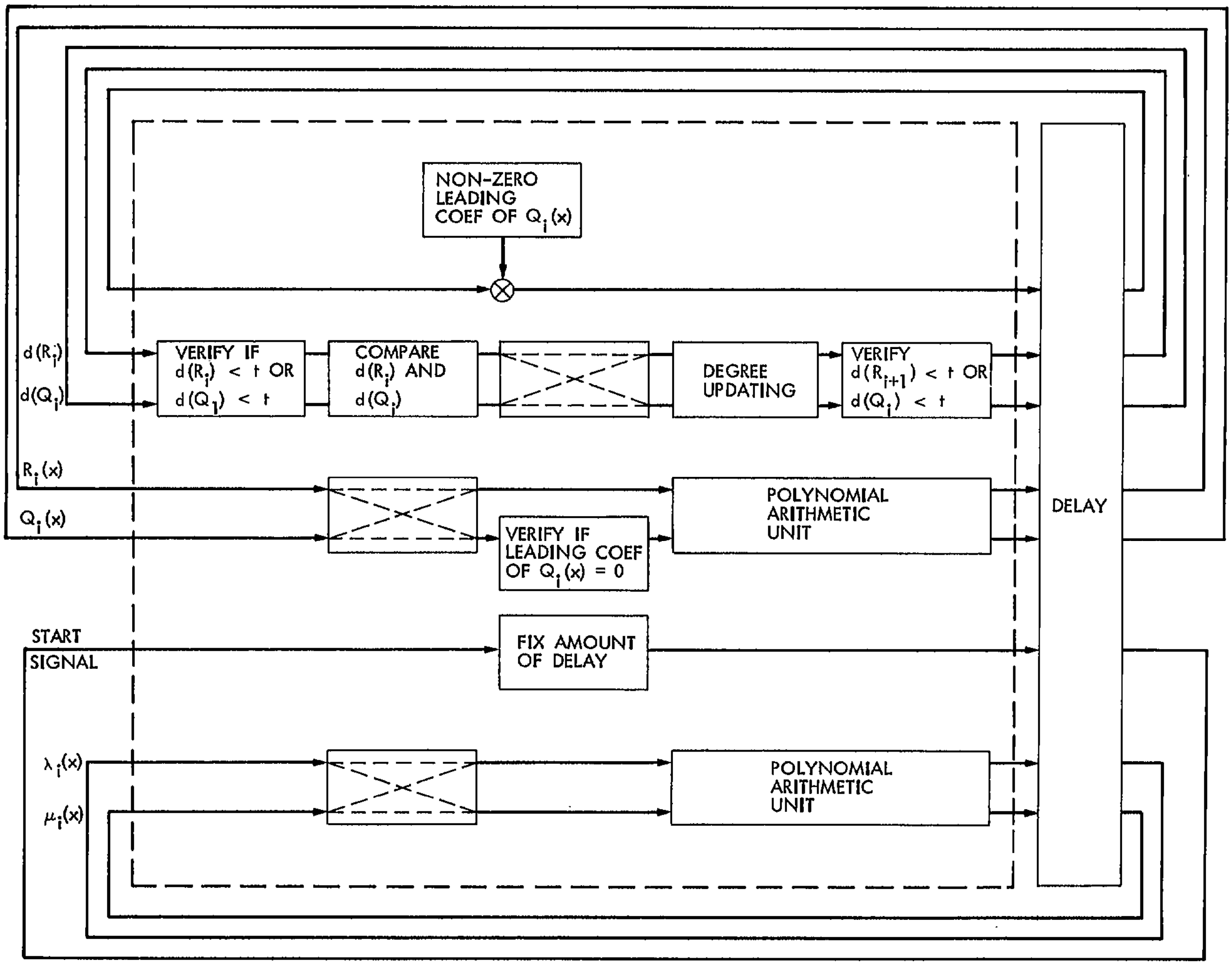
NON-ZERO LEADING COEF OF $Q_i(x)$
$d(R_i)$
$d(Q_i)$
VERIFY IF $d(R_i) < t$ OR $d(Q_i) < t$
COMPARE $d(R_i)$ AND $d(Q_i)$
DEGREE UPDATING
VERIFY $d(R_{i+1}) < t$ OR $d(Q_i) < t$
$R_i(x)$
$Q_i(x)$
VERIFY IF LEADING COEF OF $Q_i(x) = 0$
POLYNOMIAL ARITHMETIC UNIT
DELAY
START SIGNAL
FIX AMOUNT OF DELAY
$\lambda_i(x)$
$\mu_i(x)$
POLYNOMIAL ARITHMETIC UNIT

Fig. 4. The structure of a recursive cell that performs the modified Euclid's algorithm

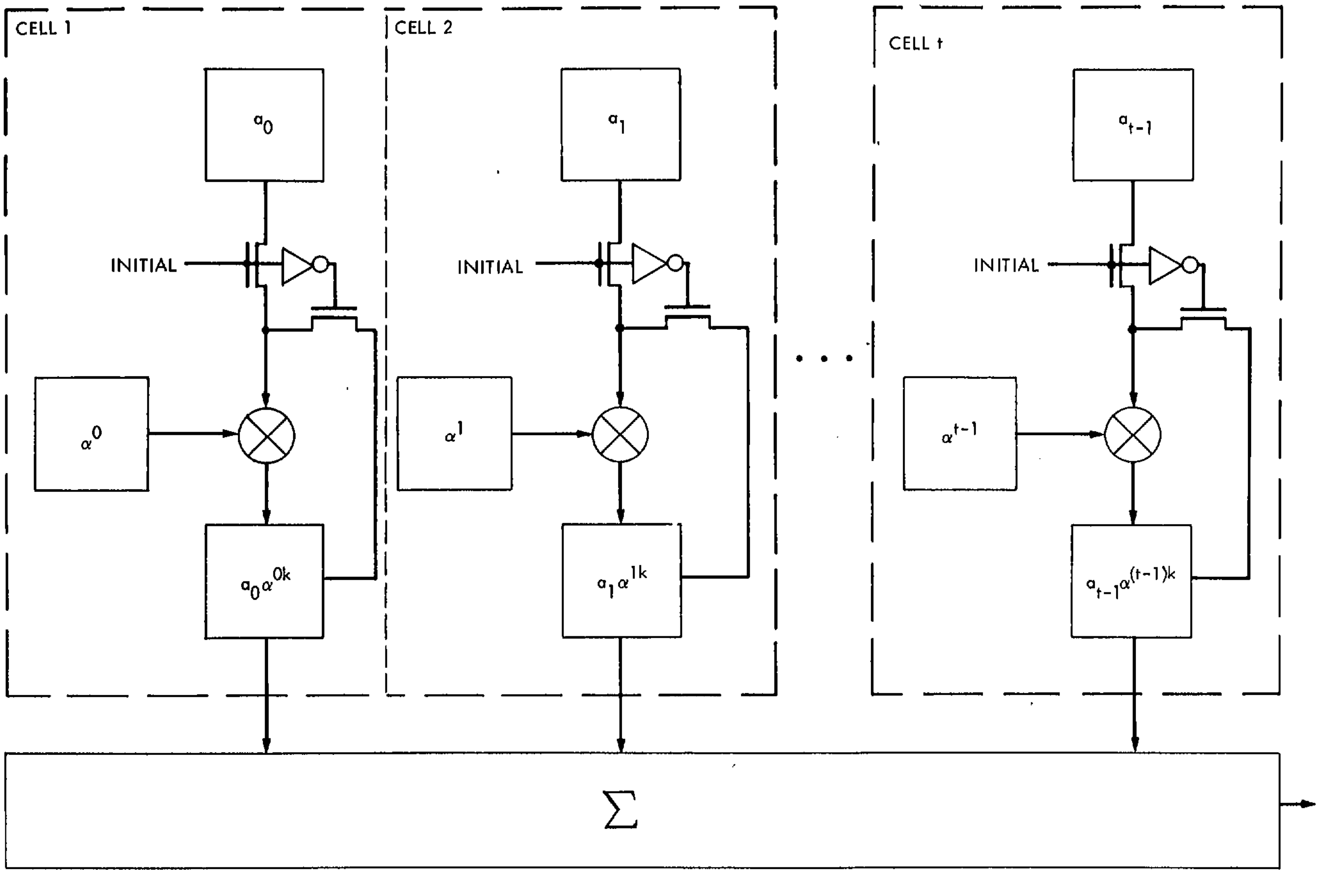

Fig. 5. A polynomial evaluation pipeline circuit