A New VLSI Architecture for a Single-Chip-Type Reed-Solomon Decoder

I. S. Hsu and T. K. Truong
Communications Systems Research Section

This article describes a new very-large-scale integration (VLSI) architecture for implementing Reed-Solomon (RS) decoders that can correct both errors and erasures. This new architecture implements a Reed-Solomon decoder by using replication of a single VLSI chip. It is anticipated that this single-chip-type RS decoder approach will save substantial development and production costs. It is estimated that reduction in cost by a factor of four is possible with this new architecture. Furthermore, this Reed-Solomon decoder is programmable between 8-bit and 10-bit symbol sizes. Therefore, both an 8-bit CCSDS RS decoder and a 10-bit decoder are obtained at the same time, and when concatenated with a (15,1/6) Viterbi decoder, provide an additional 2.1-dB coding gain.

I. Introduction

A (255,223) 8-bit Reed-Solomon (RS) code in concatenation with a (7,1/2) Viterbi-decoded convolutional code has been recommended by the Consultative Commitee for Space Data Systems (CCSDS) as a standard coding system for the DSN downlink telemetry system [1]. Figure 1 shows a CCSDS-recommended DSN transmission system. This concatenated coding system, which is the so-called standard system, provides a coding gain of about 2 dB over the (7,1/2) Viterbi-decoded-only system. In Fig. 2, several curves representing performances of different coding schemes for the DSN are illustrated [2]. Recent software simulations show that a (1023,959) Reed-Solomon code, when concatenated with a (15,1/6) Viterbi-decoded convolutional code, provide another 2-dB coding gain over the standard system recommended by CCSDS [3]. This additional coding gain may be needed for future deep-space missions to save cost, since coding is among the most cost-efficient ways to improve system performance. A VLSI-based (15,1/6) Viterbi decoder is currently being developed at JPL to support the Galileo project and is expected to be operating by mid-1991 [4]. Therefore, a (1023,959) Reed-Solomon decoder is needed to provide the remainder of the 2-dB coding gain.

Recently, several VLSI architectures for implementing Reed-Solomon decoders have been proposed [5,6]. However, the complexity of a Reed-Solomon decoder increases with the symbol size of the code. It is very unlikely that current technology can implement a single-chip Reed-Solomon decoder that can correct both errors and erasures if the symbol size of the code is larger than 8 bits. The existing VLSI Reed-Solomon decoders use a natural scheme to partition the decoder system. In this natural partitioning scheme, as many functional blocks are grouped together as possible and realized on the same

VLSI chip. For example, the VLSI chip set developed by the University of Idaho has four different types of VLSI chips [6]. The first chip computes the syndromes. The second chip is the Euclid multiply/divide unit. The third chip performs as a polynomial solver. The final chip is the error-correction chip. This kind of partitioning scheme is straightforward. However, it is expected that several different types of VLSI chips are required to implement a Reed-Solomon decoder of symbol size larger than 8 bits. The costs to design, fabricate, and test VLSI-based systems increase drastically with the number of different chip types used. The (255,223) error-correcting-only RS decoder developed by the University of Idaho [5] consists of four different types of VLSI chips. Assuming it takes eight workmonths to design and test a VLSI chip, which is a reasonable assumption for a VLSI chip of this complexity, four different chips require 32 workmonths to develop. Furthermore, assuming it costs $80,000 to fabricate a VLSI chip of this complexity, the total fabrication cost of four RS chips is $320,000. In contrast, a single-chip-type RS decoder system takes only eight workmonths to design and test, and costs $80,000. Based on the above analysis, a single-chip-type RS decoder system is expected to have a four-fold cost savings compared to RS decoder systems using conventional partition schemes.

As described above, the (255,223) 8-bit RS code has been recommended by the CCSDS as part of the standard coding scheme in the DSN telecommunication system. Software simulations show system performance improvement obtained by concatenating a (1023,959) 10-bit RS decoder with a (15,1/6) Viterbi-decoded convolutional code. Therefore, there are reasons for developing both 8-bit and 10-bit RS decoders for current and future uses, and it is desirable to realize an RS decoder that is switchable between 8-bit and 10-bit codes. The key to realizing such an RS decoder is the development of an 8-bit and 10-bit switchable finite-field multiplier, which is the most frequently used functional building block in an RS decoder.

This article describes the development of a single-chip-type Reed-Solomon decoder system that is switchable between 8-bit and 10-bit symbol sizes (although this architecture is switchable between any two symbol sizes). The VLSI architecture of this chip is described in considerable detail. The architecture is regular, simple, and expandable, and therefore relatively easy to implement and test. It is expected that RS decoder systems using this architecture will have a four-fold cost reduction compared to conventional implementation schemes.

A Reed-Solomon code is a subset of the Bose-Chaudhuri-Hocquenghem (BCH) code [7]. Therefore, a decoding technique for BCH codes can also be used to decode a Reed-Solomon code. While many schemes have been developed for decoding Reed-Solomon codes [7], the so-called "transform-domain" and "time-domain" approaches are used most frequently.

As described in [5], a transform-domain RS decoder is suitable for small symbol sizes such as 8-bit or less, while the time-domain technique is suitable for large codes such as 10-bit or more. Because of the constraints on 10-bit decoding, the time-domain approach is chosen for the design of an RS decoder which is switchable between 8 and 10 bits.

A time-domain decoding algorithm can be described in the following steps:

(1) Compute syndromes and calculate the erasure-locator polynomial.

(2) Compute the Forney syndromes.

(3) Determine the errata-locator polynomial and the errata-evaluator polynomial by applying the Euclidean algorithm.

(4) Compute the errata locations by Chien search and compute the errata values.

(5) Perform the errata corrections.

Figure 3 shows a block diagram of a time-domain RS decoder; see [5] for more details.

In Section II, the VLSI design of a programmable finite-field multiplier is described. The VLSI architecture of the single-chip-type Reed-Solomon decoder is illustrated in Section III. Finally, concluding remarks are given in Section IV.

II. The Design of a Programmable Finite-Field Multiplier

The key element in the development of a programmable 8-bit and 10-bit switchable Reed-Solomon decoder is to design an 8-bit and 10-bit programmable finite-field multiplier. Finite-field multipliers are the basic building blocks in implementing a Reed-Solomon decoder. A comparison of VLSI architectures of finite-field multipliers using dual, normal, or standard bases is discussed in [8]. Since any finite-field element can be transformed into a standard-basis representation irrespective of its original basis, this article focuses on the programmable design of a standard-basis finite-field multiplier.

Figure 4 illustrates a logic diagram of a finite-field multiplier [9]. A mathematical theory for this finite-field-multiplier architecture is described as follows [9]:

Assuming the two inputs of the multiplier are $A = \alpha^i$ and $B = \alpha^j$, respectively, where α is a primitive element of $GF(2^m)$, then A and B can be represented as

$$A = \sum_{i=0}^{m-1} a_i \alpha^i$$

$$B = \sum_{i=0}^{m-1} b_i \alpha^i$$

The product of A and B, i.e., $C = \alpha^k$, can be represented as

$$C = \sum_{i=0}^{m-1} c_i \alpha^i$$

By the use of Horner's rule, the product C can be written

$$C = AB = A \sum_{k=0}^{m-1} b_k \alpha^k$$

$$= (\ldots((Ab_{m-1}\alpha + Ab_{m-2})\alpha + Ab_{m-3})\alpha$$

$$+ \ldots Ab_1)\alpha + Ab_0$$

or

$$C^{(0)} = Ab_{m-1}$$

$$C^{(1)} = Ab_{m-1}\alpha + Ab_{m-2} = C^{(0)}\alpha + Ab_{m-2}$$

$$C^{(i)} = C^{(i-1)}\alpha + Ab_{m-1-i}$$

$$C = C^{(m-1)} = C^{(m-2)}\alpha + Ab_0$$

Figure 5 shows the block diagram of the 10-bit standard-basis finite-field multiplier. Its extension to higher fields is obvious and straightforward. As shown in Fig. 5, this multiplier consists of 10 identical cells with each cell containing three 1-bit registers, two AND gates, and two XOR gates. There are three inputs to this multiplier. In Fig. 5, A and B represent the multiplicand and multiplier, respectively. They are represented in the basis of $\{\alpha^9, \alpha^8, \alpha^7, \alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha^1, 1\}$. Another input f in Fig. 5 is the irreducible primitive polynomial, $f(X)$, of the field. Let

$$f(X) = X^{10} + f_9X^9 + f_8X^8 + f_7X^7 + f_6X^6 + f_5X^5$$

$$+ f_4X^4 + f_3X^3 + f_2X^2 + f_1X^1 + f_0X^0$$

where $f_i \in GF(2)$. In real-world application, both A and f can be loaded into the A-register and the f-register, respectively, in either parallel or serial form. (Figure 5 shows serial form for the purpose of illustration.) However, B must come in bit-by-bit with b_9 first and b_0 last. Initially, the C-register is reset to zero. At the first clock time, $C^{(0)}$ as described above is obtained; at the second clock time, $C^{(1)}$ is obtained, and so on. After 10 clock cycles, the final product C is obtained in the C-register. It can then be shifted out either in parallel or serial form, depending on the application.

A programmable standard-basis finite-field multiplier can be easily obtained by modifying the architecture depicted in Fig. 5. Figure 6 shows an 8-bit and 10-bit programmable finite-field multiplier. When signal ET is low, representing an 8-bit version, gate G1 is off and gate G2 is on. Therefore, the feedback will be conducted at 8-bit. Of course, all three inputs to the finite-field multiplier must reformat their representation. The highest two bits, i.e., a_9, a_8, b_9, b_8, f_9, and f_8, are all set equal to zero for 8-bit operation.

III. Architecture of the Single-Chip-Type Reed-Solomon Decoder System

A. VLSI Architecture of a Single-Chip-Type Reed-Solomon Decoder

This section describes the VLSI architecture of a single-chip-type Reed-Solomon decoder. The development of this new architecture is based on the VLSI architecture of an RS decoder described in [5]. Because of the regularity of a time-domain RS decoder structure, the functional units in an RS decoder can be efficiently partitioned. Figure 7 shows the architecture of a VLSI chip that is a basic building block of the single-chip-type Reed-Solomon decoder system. As shown in Fig. 7, the VLSI chip is partitioned into six rows. The first row of the chip consists of eight identical syndrome subcells. The 8-bit or 10-bit RS decoder is realized by making both the shift registers and the finite-field multipliers in all the subcells programmable between 8-bit and 10-bit operation.

The second row of the chip has eight polynomial expansion subcells; the third row consists of eight power expansion subcells. The fourth row of the chip has eight polynomial evaluation subcells which can also be used to do the Chien search operation. The fifth row has eight modified Euclidean subcells. Finally, the sixth row of the VLSI chip contains some miscel-

laneous cells such as counters, shift registers, finite-field multipliers and so forth. These miscellaneous cells are used as glue logic in a VLSI RS decoder system. As shown in Fig. 8, if four of these VLSI chips are connected in an array, a (255,223) time-domain RS decoder is formed since there are enough subcells to implement all the functional units. In other words, there are 32 syndrome subcells, 32 polynomial expansion subcells, 32 power expansion subcells, 32 polynomial evaluation/Chien search subcells, and 32 modified Euclidean algorithm subcells. Since all the subcells are programmable between 8-bit and 10-bit, the core of a 10-bit (1023,959) RS decoder is formed by arraying eight copies of this VLSI chip.

It is estimated that the total number of pins required for a VLSI chip is less than 132 and the total number of transistors per chip is less than 60,000. Obviously, these requirements are within today's VLSI technology capability.

The number of subcells in a VLSI chip could be reduced by half to decrease the silicon real estate and therefore increase chip yield. That is, only four subcells in each functional unit would be implemented on a VLSI chip. The number of transistors is reduced from 60,000 to 30,000 by this arrangement. On the other hand, if good fabrication technology is available, the number of functional subcells in a chip could be doubled such that the chip count in an RS decoder system is reduced by half. Therefore, this RS decoder architecture provides the maximum flexibility in both the chip and system designs.

B. Configuration of a Single-Chip-Type RS Decoder System

The system configuration of the proposed Reed-Solomon decoder is depicted in Fig. 9. As shown in Fig. 9, the system is partitioned into five units. There is a host computer (which could be a personal computer) to issue commands to the whole system. An input module which consists mostly of memory chips is used to store the received messages. Operations such as formatting, basis conversion if both standard and dual bases are used, zero-fill, etc., will be performed in this unit. Similarly, an output module is used to store the decoded symbols and perform operations such as basis reconversion, reformatting, and zero-stripping.

A control memory unit is used to store all the control signals for the VLSI chip. Due to the large number of control signals required for VLSI chips, it is not effective to include the control signal generation unit in the VLSI chip. The partitioning of the VLSI chip becomes very difficult if the control signal generators are included. It is expected that the control memory unit will consist of EPROMs which store control signals for the VLSI chip. Further modifications or expansions of control signals for the VLSI chip will be relatively easy in this scheme. Finally, the fifth part of the RS decoder system is the RS decoder VLSI chip set. This is the core of an RS decoder system.

IV. Conclusion

This article describes a new architecture for implementing a Reed-Solomon decoder. This new architecture uses a single-chip-type scheme that provides a minimum four-fold cost savings when compared to other RS decoder implementations. It is shown that a programmable finite-field multiplier will realize an 8-bit and 10-bit switchable RS decoder. The system configuration of an RS decoder is also described. An array of four identical VLSI chips forms an 8-bit CCSDS RS decoder and an array of eight identical VLSI chips forms a 10-bit RS decoder.

References

[1] "Recommendation for Space Data System Standards: Telemetry Channel Coding," (Blue Book), Consultative Committee for Space Data Systems, CCSDS Secretariat, Communications and Data Systems Division, Code TS, NASA, Washington, D.C., May 1984.

[2] R. L. Miller, L. J. Deutsch, and S. A. Butman, *On the Error Statistics of Viterbi Decoding and the Performance of Concatenated Codes*, JPL Publication 81-9, Jet Propulsion Laboratory, Pasadena, California, September 1, 1981.

[3] J. H. Yuen and Q. D. Vo, "In Search of a 2-dB Coding Gain," *TDA Progress Report 42-83*, vol. July–September 1985, Jet Propulsion Laboratory, Pasadena, California, pp. 26–33, November 15, 1985.

[4] J. Statman, G. Zimmerman, F. Pollara, and O. Collins, "A Long Constraint Length VLSI Viterbi Decoder for the DSN," *TDA Progress Report 42-95*, vol. July–September 1988, Jet Propulsion Laboratory, Pasadena, California, pp. 134–142, November 15, 1988.

[5] I. S. Hsu, T. K. Truong, I. S. Reed, L. J. Deutsch, and E. H. Satorius, "A Comparison of VLSI Architectures for Time and Transform Domain Decoding of Reed-Solomon Codes," *TDA Progress Report 42-92*, vol. October–December 1987, Jet Propulsion Laboratory, Pasadena, California, pp. 63–81, February 15, 1988.

[6] G. K. Maki, P. A. Owsley, K. B. Cameron, and J. Venbrux, "VLSI Reed-Solomon Decoder Design," *Proceedings of the Military Communications Conference (MILCOM)*, Monterey, California, pp. 46.5.1–46.5.6, October 5–9, 1986.

[7] R. E. Blahut, *Theory and Practice of Error Control Codes*, Reading, Massachusetts: Addison-Wesley, May 1984.

[8] I. S. Hsu, T. K. Truong, I. S. Reed, and L. J. Deutsch, "A Comparison of VLSI Architectures of Finite Field Multipliers Using Dual, Normal or Standard Bases," *IEEE Trans. on Computers*, vol. 37, no. 6, pp. 735–739, June 1988.

[9] P. A. Scott, S. E. Tavares, and L. E. Peppard, "A Fast VLSI Multiplier for $GF(2^m)$," *IEEE Journal on Selected Areas in Communications*, vol. SAC-4, no. 1, pp. 62–66, January 1986.

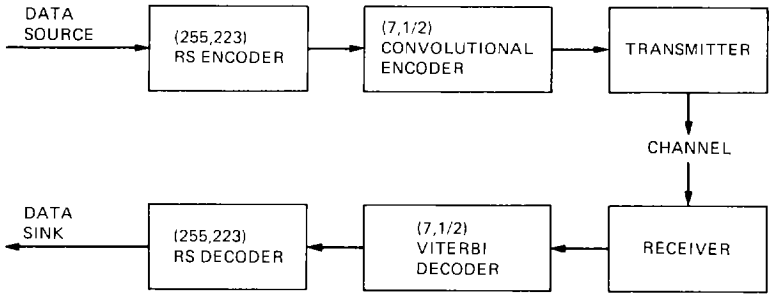

Fig. 1. Standard DSN telemetry coding system recommended by CCSDS.

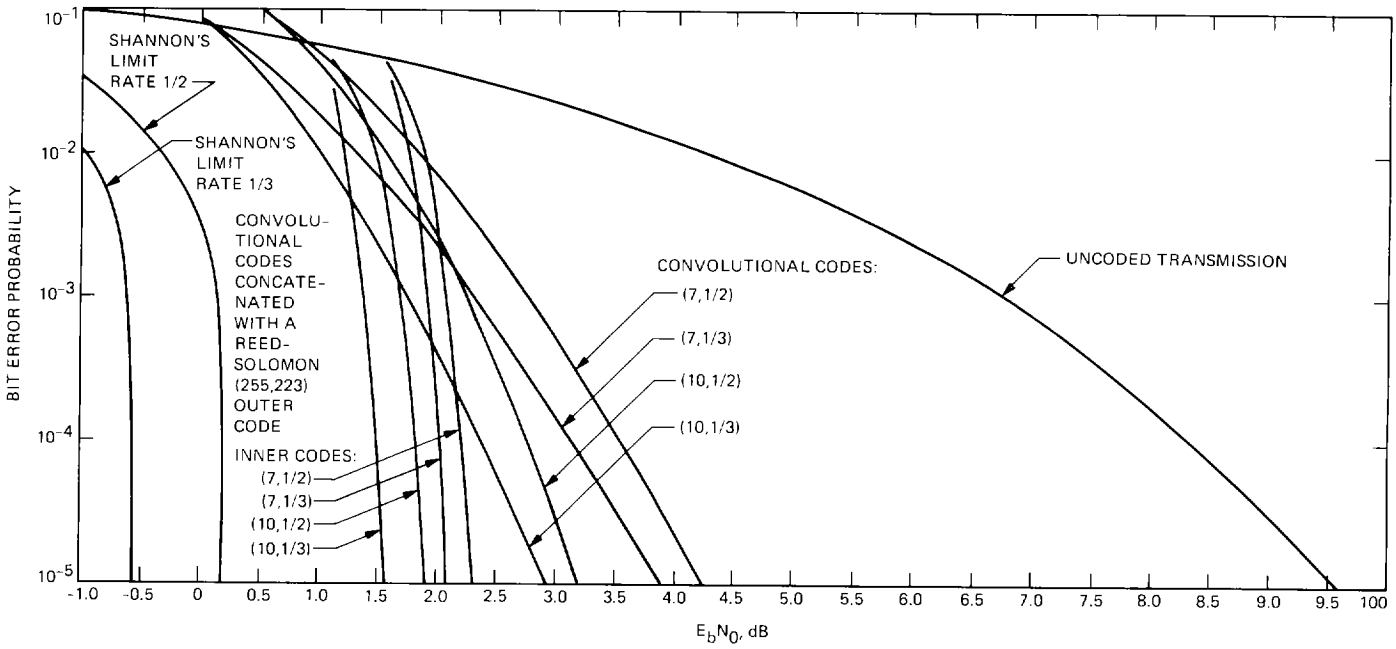

Fig. 2. Performance curves of various coding schemes in [2].

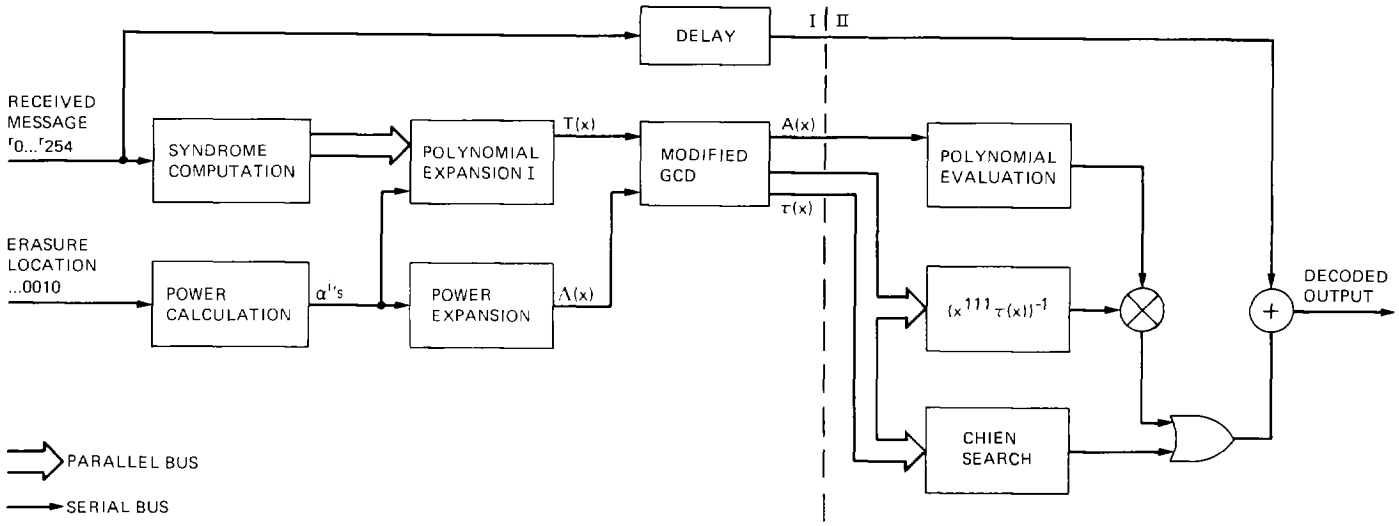

Fig. 3. Block diagram of a time-domain RS decoder for errors and erasures.

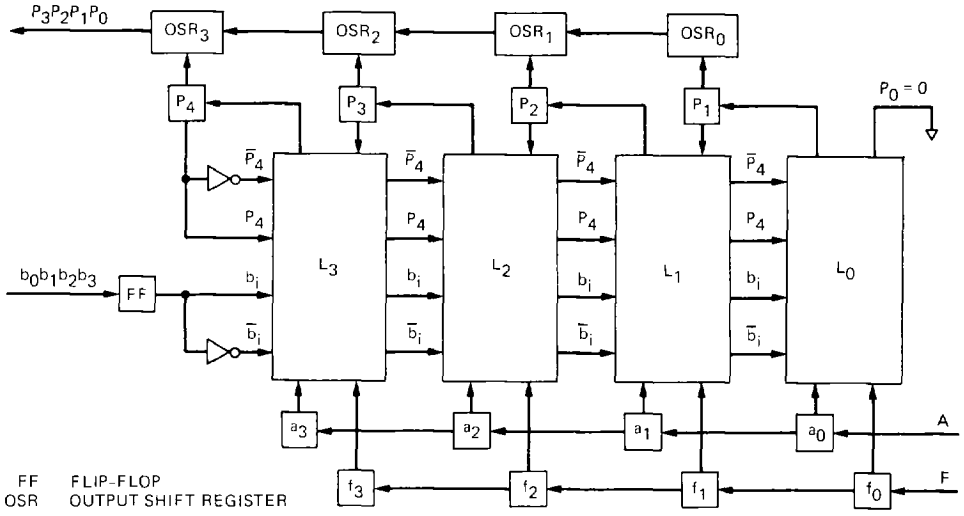

Fig. 4. Logic diagram of the standard-basis finite-field multiplier.

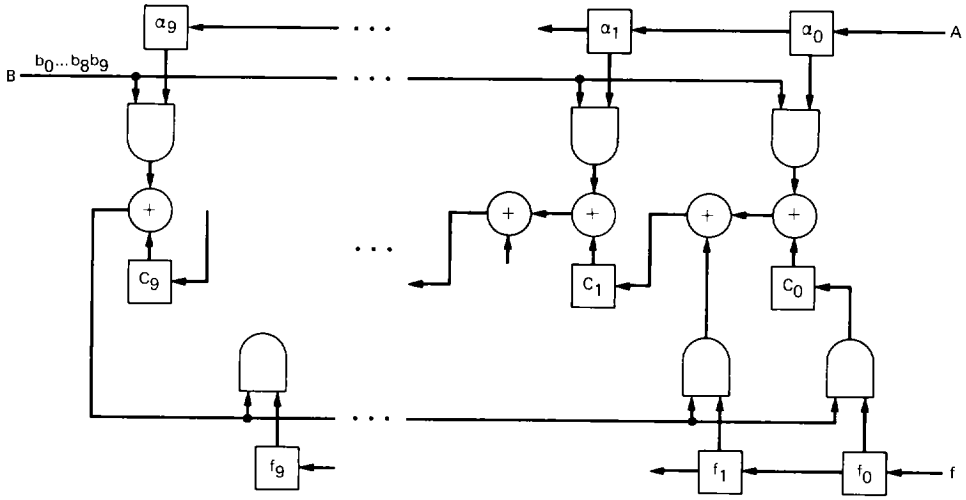

Fig. 5. Block diagram of a 10-bit standard-basis finite-field multiplier.

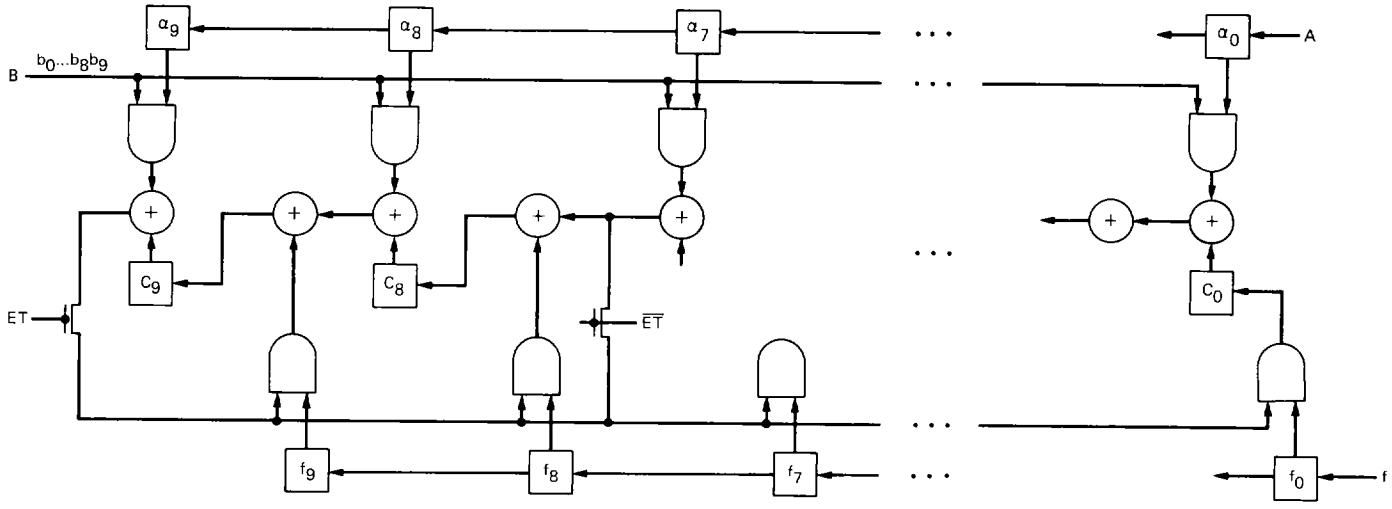

Fig. 6. Block diagram of an 8-bit and 10-bit switchable standard-basis finite-field multiplier.

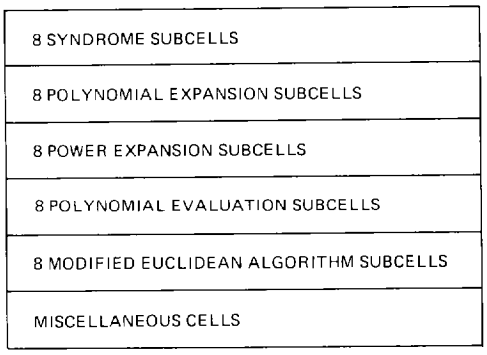

Fig. 7. Block diagram of VLSI chip architecture in the single-chip-type RS decoder system.

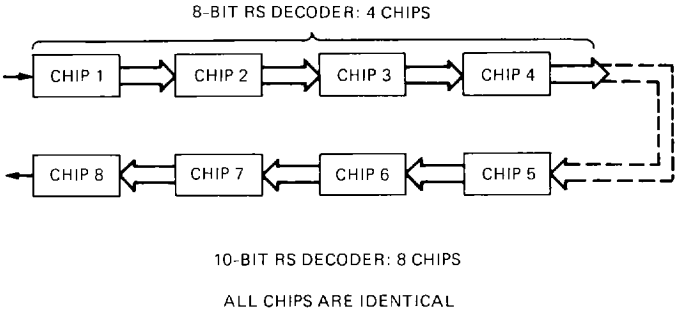

Fig. 8. Architecture of the 8-bit and 10-bit switchable RS decoder system.

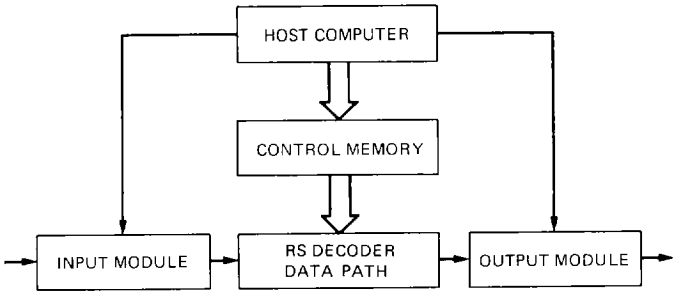

Fig. 9. Configuration of the proposed RS decoder system.