Architecture for VLSI Design of Reed-Solomon Encoders*

K.Y. Liu
Jet Propulsion Laboratory, Pasadena, CA. 91109

Abstract

In this paper, the logic structure of a universal VLSI chip called the symbol-slice Reed-Solomon (RS) encoder chip is presented. An RS encoder can be constructed by cascading and properly interconnecting a group of such VLSI chips. As a design example, it is shown that a (255,223) RS encoder requiring around 40 discrete CMOS IC's may be replaced by an RS encoder consisting of four identical interconnected VLSI RS encoder chips. Besides the size advantage, the VLSI RS encoder also has the potential advantages of requiring less power and having a higher reliability.

I. Introduction

Reed-Solomon (RS) codes [1] are non-binary BCH codes. These codes can correct both random and burst errors over a communication channel. Recently concatenated coding systems using RS codes as the outer codes have been proposed for space communication to achieve very low error probabilities [2]-[7]. Several deep space flight projects such as the Voyager at Uranus encounter, the Galileo, and the International Solar Polar Mission (ISPM) have also considered using the concatenated RS/Viterbi channel coding scheme. Hence RS codes are quite important for space communications.

The complexity of an RS encoder is proportional to the error-correcting capability of the code, the speed of the encoding, and the interleaving level used [4]. For reliable space communication there is a need to use RS codes

*This paper presents one phase of research conducted at the Jet Propulsion Laboratory, California Institute of Technology under contract No. NAS-7-100 sponsored by the National Aeronautics and Space Administration.

with large error-correcting capability and large interleaving level [4], [5], [8]. Hence one is especially interested in minimizing the complexity of RS encoders for space communication applications. In a spacecraft the power, size, and reliability requirements are usually quite severe. Thus there is considerable interest in a VLSI (Very Large Scale Integration) RS encoder which has the potential for significant savings in size, weight, and power while at the same time providing higher reliability over an RS encoder implemented in discrete logic circuits.

This paper introduces a symbol-sliced logic structure suitable for a VLSI implementation of RS encoders. By cascading and properly interconnecting a group of such VLSI chips, each consisting of a fixed portion of the encoder, it is possible to obtain an RS encoder with any desired error-correcting capability and interleaving level. As a design example, it is shown that a (255,223) RS encoder requiring 40 discrete CMOS IC's may be replaced by an RS encoder consisting of four identical interconnected VLSI encoder chips.

II. Reed-Solomon Encoding Procedures

An RS codeword has (2^J-1) symbols, where each symbol has J bits. Of the (2^J-1) symbols there are (2^J-1-2E) information symbols and 2E parity-check symbols, where E is the number of symbols an RS code is able to correct. If one treats the (2^J-1-2E) information symbols as the coefficients of the polynomial

$$f(x) = x^{2E}\left(s_{2^J-1-2E} + s_{2^J-2-2E}x + \cdots + s_2 x^{2^J-2-2E} + s_1 x^{2^J-1-2E}\right)$$

where s_i is the ith transmitted symbol, then the 2E parity-check symbols can be obtained as the coefficients of the remainder of

$$f(x)/g(x)$$

where g(x) is the generator polynomial [9] of the code. Usually g(x) is defined as

$$g(x) = \prod_{i=1}^{2E} (x-\alpha^i) = \sum_{j=o}^{2E} g_j \, x^j$$

where α is a primitive element of the Galois field $GF(2^J)$, and g_j's are the coefficients of $g(x)$ with $g_{2E} = 1$. The generator polynomial defined above does not have symmetrical coefficients, i.e.,

$$g_j \neq g_{2E-j} \text{ for } j = 0, 1, 2, \text{----}, 2E.$$

A block diagram of an RS encoder which generates the remainder of $f(x)/g(x)$ is given in Fig. 1. The switches in Fig. 1 are normally in the "ON" position until the last information symbol gets into the encoder. At this moment all switches are switched to the "OFF" position and the encoder is behaving like a long shift register. The output of the encoder is then taken from the output of the last shift register. Note that in Fig. 1 2E multipliers are needed in the encoder.

To reduce the number of multipliers needed, a special class of the generator polynomial which has symmetrical coefficients was proposed by Berlekamp [10]. This generator polynomial is defined as

$$g(x) = \prod_{i=2^{J-1}-E}^{2^{J-1}+E-1} (x-\alpha^i) = \sum_{j=o}^{2E} g_j x^j$$

where

$$g_j = g_{2E-j} \text{ and } g_o = g_{2E} = 1.$$

Note that since $g_o = 1$, only E multipliers are needed. Thus using this new generator polynomial will reduce the number of multipliers required by one-half.

There are several schemes for interleaving the RS codes [5]. One scheme illustrated in Fig. 3 as "Interleave B" requires memory only for the parity-check symbols in the encoder is described as follows. In this scheme the input bits are grouped into J-bit symbols and transmitted in their natural order. However every I^{th} symbol belongs to the same codeword, where I is

the interleaving depth used. Thus I codewords make up such an interleaved code block. After the information symbols are transmitted, the parity-check symbols of each interleaved codeword are then transmitted.

If interleaving is used, then the encoder logic structure is the same as shown in Fig. 1, except now each J-bit shift register is replaced by an I x J - bit shift register. As an example, a block diagram of a (255,223) RS encoder with interleaving level I and generator polynomial

$$g(x) = \prod_{i=112}^{143} (x-\alpha^i)$$

where $\alpha = 2$ in $GF(2^8)$, which is generated by the primitive polynomial

$$x^8 + x^4 + x^3 + x^2 + 1,$$

is shown in Figure 4. Note that a generator polynomial with symmetrical coefficients is used here to save multipliers.

III. Symbol-Slice VLSI RS Encoder Architecture

A finite field multiplication is a quite complicated operation. There are basically three techniques for implementing a finite field multiplication. The first technique is to use log and antilog tables stored in read-only memories (ROM's) [4]. The second technique is to use a linear feedback shift register type of approach [9]. The third technique is to use the property of the trace in a finite field to form a smaller ROM look-up table [10]. Due to the advent of LSI ROM technology, techniques 1 and 3 are usually used in an RS encoder design optimized for discrete IC's. As an example a 400 KHZ (255,233) RS encoder using the Berlekamp's approach [10] requires only around 40 CMOS IC's.

When one is interested in further drastic reduction of the power and size of an RS encoder for high speed applications, one has to consider VLSI implementations. An RS encoder design optimized for discrete IC's usually does not have a modular structure. Hence when one uses such a architecture for VLSI layout, one has the following problems:

(1) The design is too big to be put on one chip,

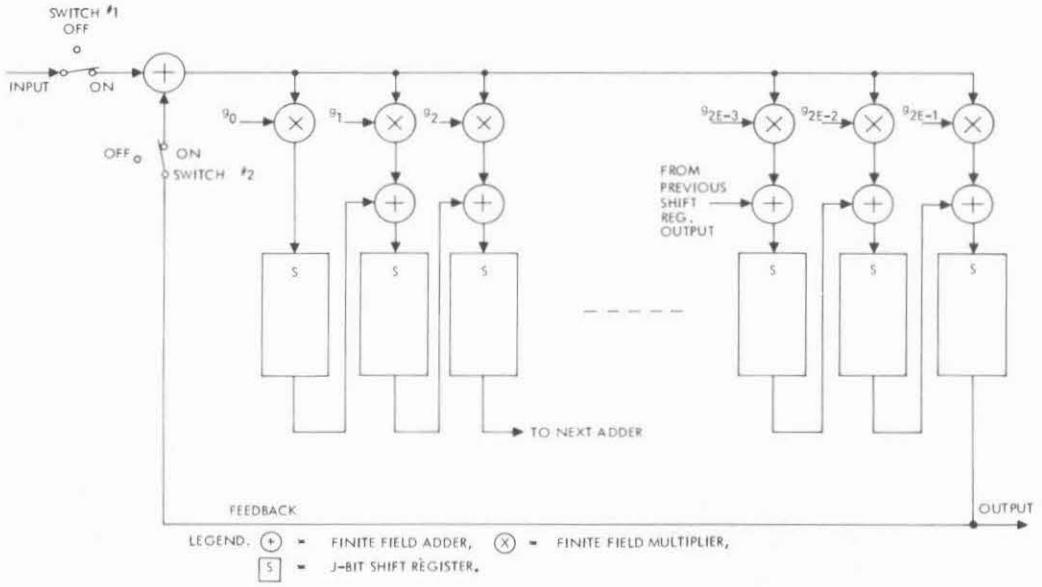

Figure 1. A Block Diagram of a $(2^J-1, 2^J-1-2E)$ RS Encoder Using a Conventional Generator Polynomial.

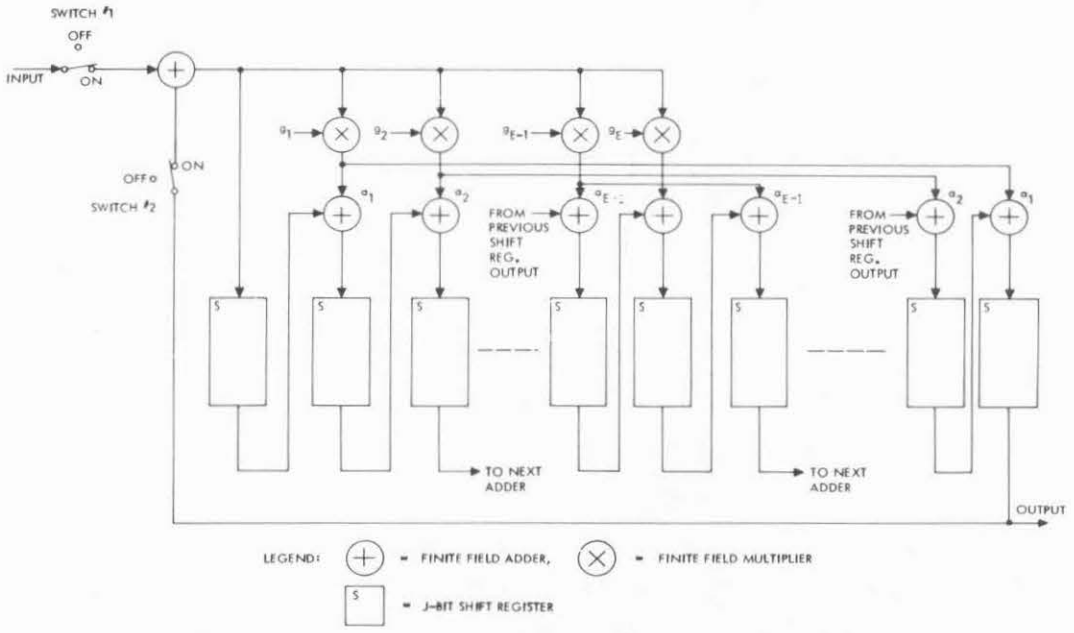

Figure 2. A Block Diagram of a $(2^J-1, 2^J-1-2E)$ RS Encoder Using a Generator Polynomial With Symmetrial Coefficients.

(2) If a multiple-chip approach is used, then one needs several chip designs, where each chip has an impractical number of input/output pins.

(3) The design is not modular. Therefore the design is not easy to adapt to other RS code parameters.

Hence there is a need to find a VLSI logic structure which can alleviate the above problems.

The repetitive architecture of the RS encoders shown in Figs. 1, 2, and 4, suggested that a symbol-slice type of VLSI chips, each one consisting of a fixed portion of the encoder, may be cascaded to form a complete RS encoder. Also to reduce the VLSI chip size, RS encoders using generator polynomials with symmetrical coefficients are preferred. Hence we will put emphasis on this type of VLSI RS encoder.

As an example, we will design a VLSI encoder chip for a 255-symbol, 8-bit per symbol, 16-error-correcting, RS code with an interleaving level of 5. The primitive polynomial used is

$$x^8 + x^4 + x^3 + x^2 + 1$$

The generator polynomial for each codeword is

$$g(x) = \prod_{i=112}^{143} (x-\alpha^i)$$

where $\alpha = 2$. The encoder logic structure for the above RS code parameter is identical to the one shown in Fig. 4, except now I = 5. There are several ways of partitioning the RS encoder into four sections. One way which requires a minimum of input/output pins is to include four rows of logic shown in Fig. 4 into one section. Each section is then realized by a universal VLSI RS encoder chip. The logic structure of this universal chip is shown in Fig. 5. The entire VLSI encoder system, which consists of four identical VLSI RS encoder chips cascaded and properly interconnected together is shown in Fig. 6. Each VLSI RS encoder chip has 24 pins. A detailed description of the VLSI RS encoder chip and the entire VLSI RS encoder system is described as follows:

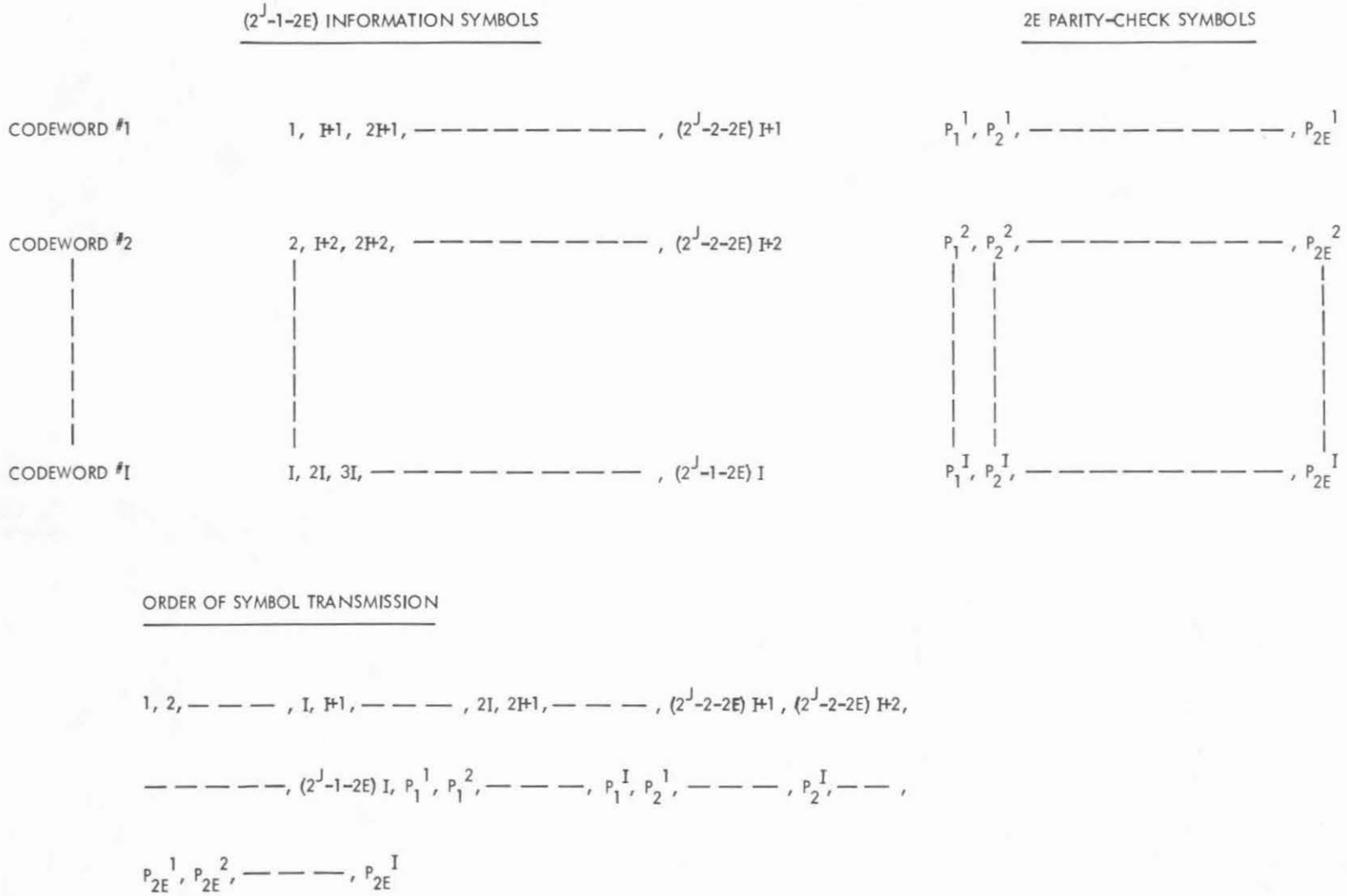

Figure 3. Code Array Structure and Order of Symbol Transmission For Type B Interleaving, Where Interleaving, Where Interleaving Level = I

3.1 Generator Polynomial Coefficients Table

Since a generator polynomial g(x) with symmetrical coefficients is used, the coefficients of x^0 is always 1. Hence there is no need for a multiplier to operate on this coefficient (see Fig. 4). Consquently if the new generator polynomial is used, then one only needs E/N multipliers on each VLSI chip, where E is the error correcting capability of the code and N is the total number of chips required in a VLSI encoder system. In the design example, E = 16 and N = 4. Hence 4 multipliers are used on each VLSI RS encoder chip. To make the VLSI chip universal, all distinct coefficients except 1 of the generator polynomial are stored in a read-only memory on the chip. In general, an ExJ-bit table is needed. In the design example E = 16, J = 8. Hence a 16 x 8-bit table is selected. The outputs of an ExJ-bit table is fed into N, E/N-to-one multiplexors. The outputs of the multiplexors are selected by $\log_2 N$ input pins called the "chip select" or "G select" pins. These outputs are then fed into the inputs of the E/N multipliers. In the design example two "G select" pins (pins 22 and 23) and four, four-to-one multiplexors are used. The 16 x 8 table and the multiplexors can easily be implemented by four, 4 x 8 ROM, with G select signals as the address control lines of each ROM. The coefficients of g(x) selected on each chip are shown in Fig. 6.

3.2 Finite Field Multiplier

Next we will discuss the architecture of the finite field multiplier. To connect the multiplier properly between chips and at the same time minimize the number of input/output pins used, a linear feedback shift register type of multiplier [9] rather than a ROM table lookup type of multiplier [4] is adopted. The multiplier used is of a serial-parallel type. The J-bit generator polynomial coefficient is read out from the ExJ-bit ROM table and fed into the multiplier J-bit in parallel whereas the other input, generated by the feedback input (pin 2) "ANDed" with the feedback enable (pin 1), is fed into the multiplier bit-by-bit in serial.

The output of the multiplier is loaded into an 8-bit shift register in parallel at the end of every 8th bit clock (1 symbol clock time). The parallel data is serialized by this shift register. The most significant bit (MSB) output of this shift register is added with the MSB of the 40-bit shift

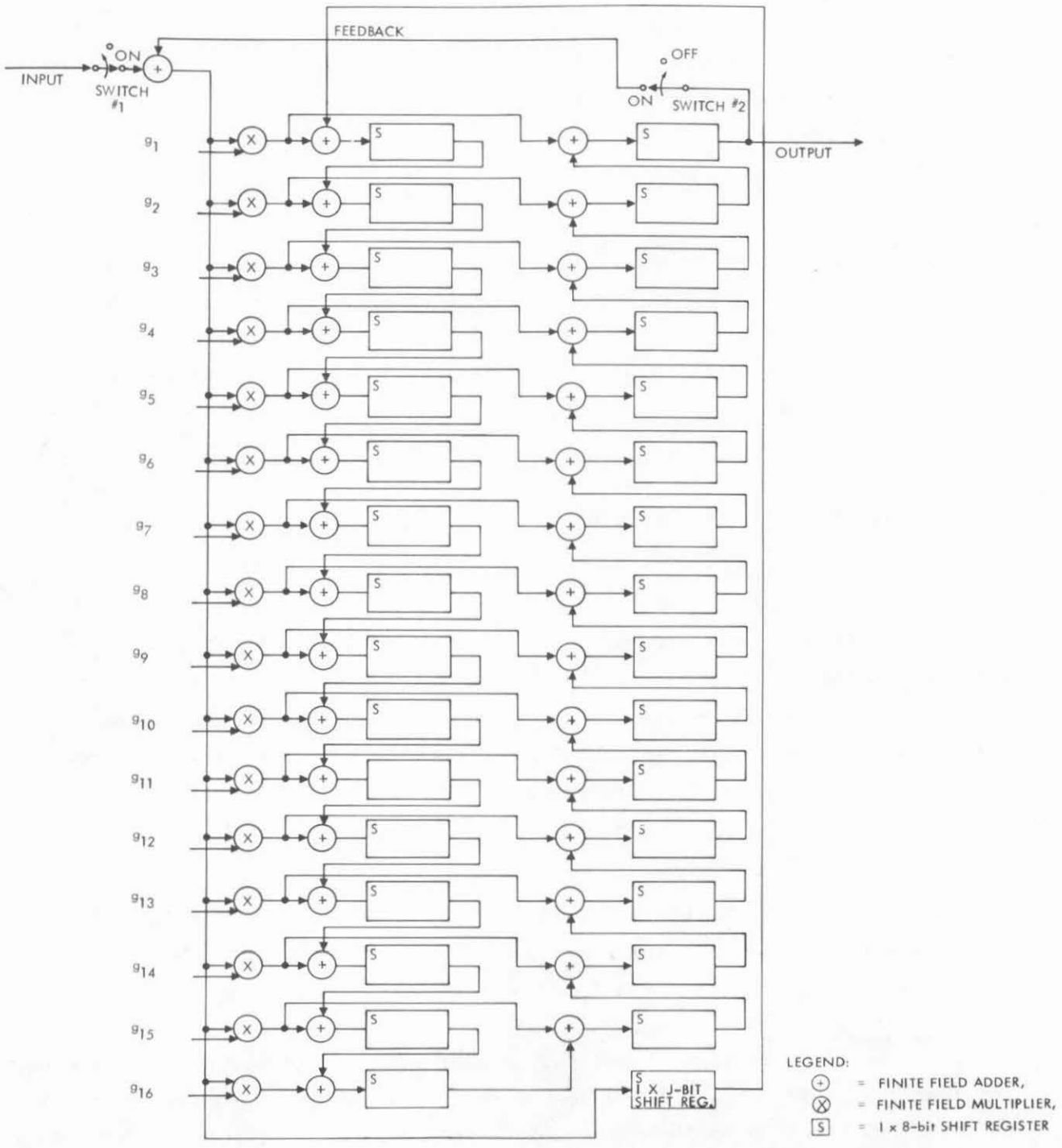

Figure 4. A Block Diagram of a (255,223) RS Encoder with Interleaving Level I and New Generator Polynomial.

register, which is either on the same chip or on a different chip, and the resulting data is shifted into the least significant bit (LSB) of the next 40-bit shift register. The adder is implemented by a two-input EXCLUSIVE-OR gate (there are eight EXCLUSIVE-OR gates on each chip). Each adder takes an input from a multiplier output which is either on the same chip or on a different chip, depending on the coefficients of the generator polynomial.

3.3 Input and Feedback Control Switches

The switch #1 shown in Fig. 4 is implemented by an AND gate on the chip with the bit-serial data input (pin 3) and the feedback enable signal as the two inputs. The feedback enable signal is provided by an external modulo 255 counter which is driven by a clock equal to the bit clock divided by 40 (see Fig. 6). This signal is true when the counter is counting from 1 to 223; otherwise it is false. The output of switch #1 is added with the MSB of the 40-bit shift register S5 output on the same chip to generate the feedback output signal (pin 5). This signal is redundant in all but the first chip (see Fig. 6).

The switch #2 shown in Fig. 4 is implemented by an AND gate on the chip with the feedback enable and feedback input signals as the two inputs. The feedback input signals on all chips (pin 2) are connected to the feedback output signal (pin 5) on the first VLSI chip. The output of switch #2 is fed into all multipliers on the chip bit-by-bit in serial.

3.4 Input/Output Data Connections

There are eight input/output lines on each chip. Of these eight lines, four lines (pins 8, 9, 11, 17) are input lines and the remaining are output lines (pins 6, 7, 10, 13). Pin 8 is normally connected to pin 6 on the same chip except for the last chip, where pin 8 is grounded. Pins 7 and 9 are normally connected to pins 17 and 13, respectively on the next chip except for the last chip, where pin 7 is connected to pin 11 on the same chip and pin 9 is connected to pin 4 on the first chip. Thus one has a railroad type of data connections between chips. The reason for connecting pin 4 on the first chip to pin 9 on the last chip is a consequence of an inherent 8-bit multiplier delay. To replace the multiplier on the x^0 position by the switch #1 output,

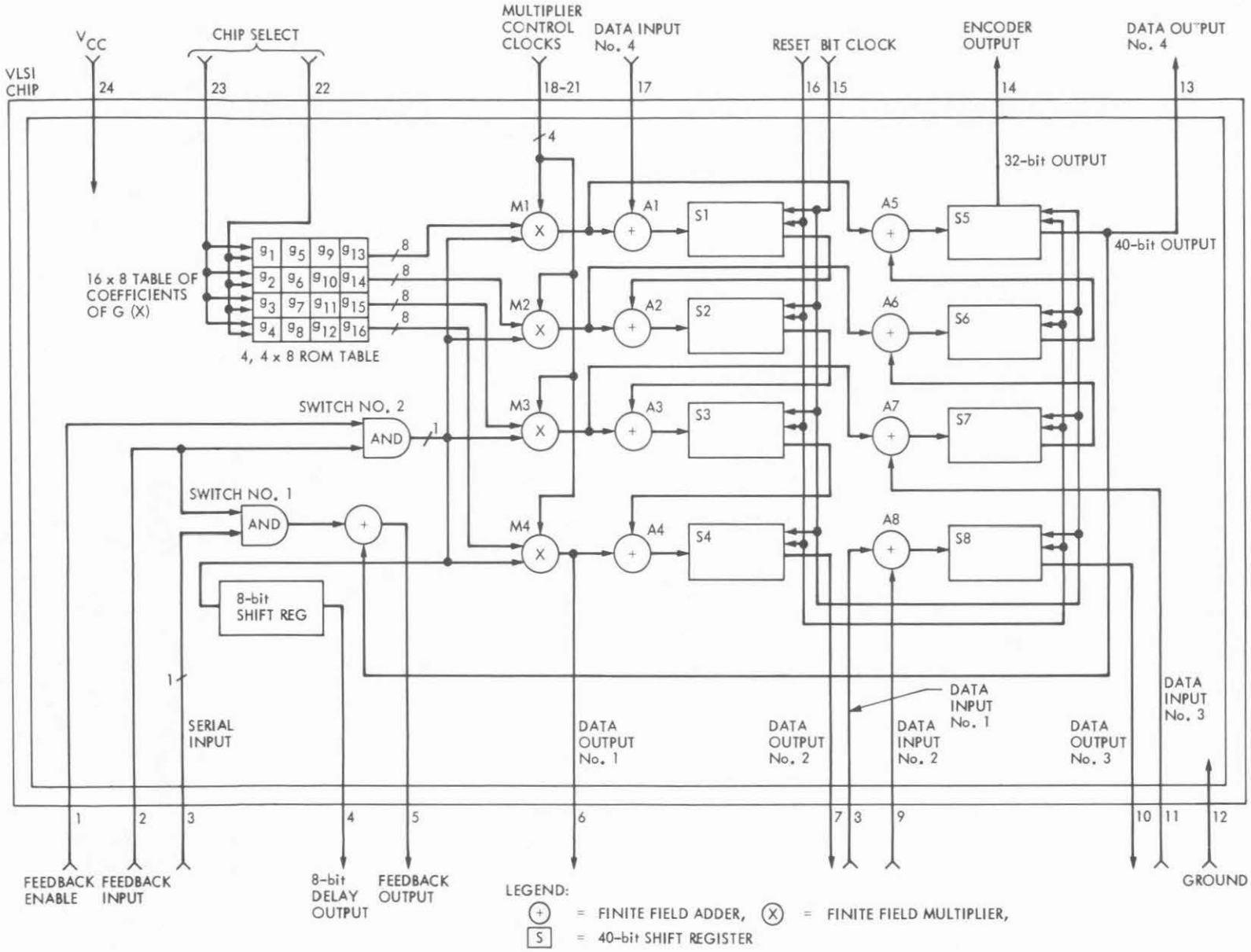

Figure 5. VLSI RS Encoder Chip Logic Structure

one needs to delay this output also by 8 bits to lineup the bits. For other chips besides the first chip, the 8-bit register outputs are not used.

The encoder output is taken 8 bits earlier from the MSB of the last 40-bit shift register on the first chip. This is because when the last bit of the last symbol in the information part of the code is shifted into the encoder, the contents of each multiplier are loaded into the 8-bit output shift registers waiting to be added with the MSB of the 40-bit shift registers. These 8-bit symbols in the multiplier output registers are actually belong to the fifth codeword in the interleaved code array. However the parity-check of the first codeword is already being computed and now sitting 8 bits from the MSB's of each 40-bit shift registers. Hence the 32-bit output (pin 14) of the last 40-bit shift register on the last chip is the output of the VLSI RS encoder system. This output is taken when the external modulo 255 counter is counting from 224 to 255. Since at these times switches #1 and #2 are turned off, the entire VLSI encoder system is behaving like an Ix2ExJ-bit (e.g. 5 x 32 x 8-bit in the design example) shift register. Thus the 5 x 32, 8-bit parity-check symbols are read out from the VLSI RS encoder bit-by-bit in serial and appended to the 5 x 223, 8-bit information symbols.

Another way to partition the RS encoder shown in Fig. 4 into four sections is to include 8 rows of logic in each column into one section. Each section is then realized by a universal VLSI RS encoder chip. The logic structure of this chip is very similar to the one shown in Fig. 5 except now one has to provide output pins to all multipliers and input pins to all adders on the chip. Hence this chip uses 6 more input/output pins than the one shown in Fig. 5. The interchip connections between adders and multipliers are similar to the pyramid type of connections shown in Fig. 2.

IV. VLSI RS Encoder Chip Size Assessment

It is estimated that it is impossible to put the entire VLSI RS encoder chip design on a 235x235 mils CMOS/bulk VLSI chip using a 7μ standard cell approach on all logic. However, if one uses custom RAM and ROM cells design to implement the 40-bit static shift registers and the 16x8 table and 7μ standard cell design for the rest of the logic, then it is possible to have a one-chip design.

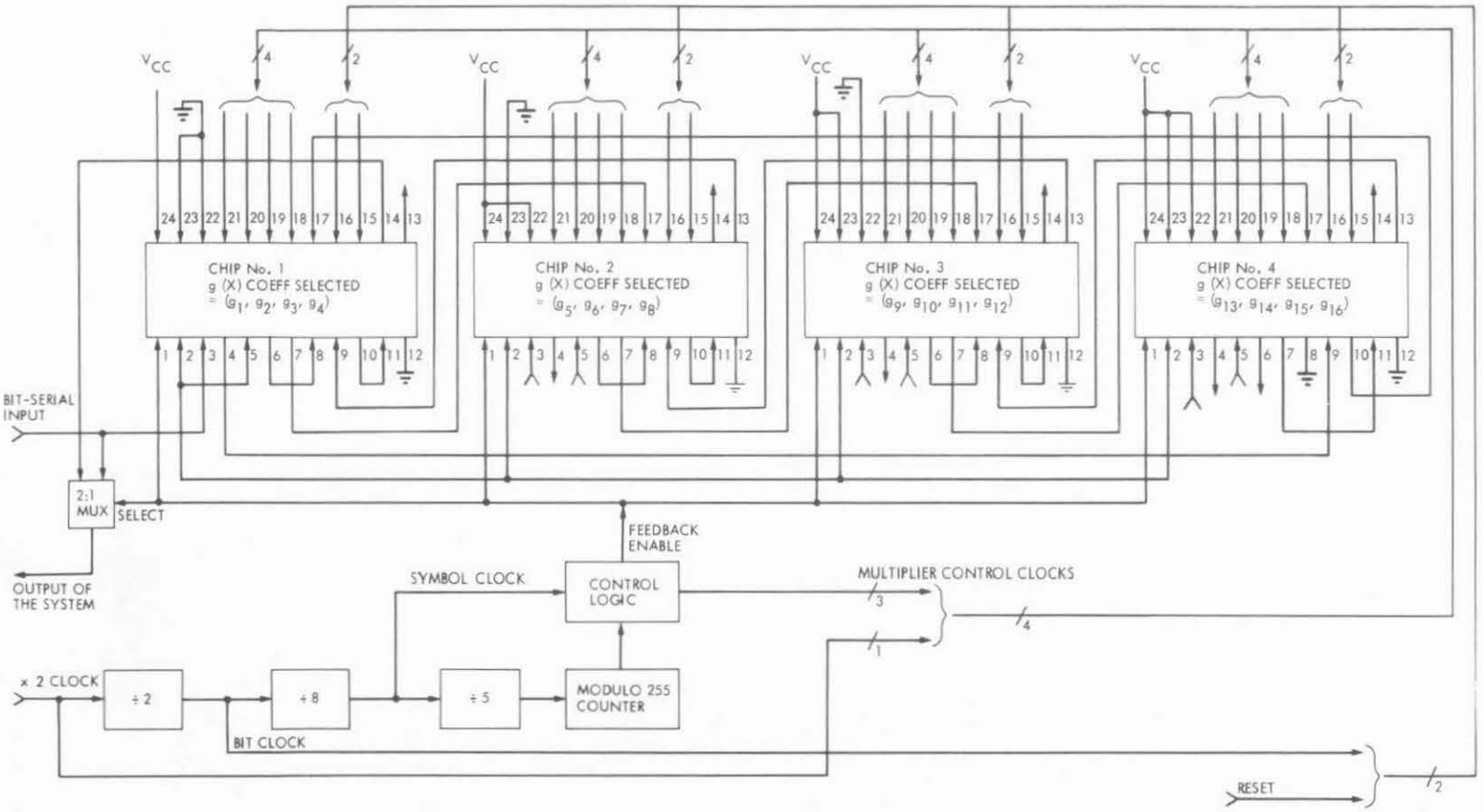

Figure 6. VLSI RS Encoder System Diagram

V. Performance of the VLSI RS Encoder System

For design verification, both the shift register version and the RAM version of the VLSI RS encoder chip and VLSI RS encoder system are implemented using discrete CMOS IC's and are now operational. The throughputs of these two versions are 800K bits/sec for the shift register version and 200K bits/sec for the RAM version. These throughputs are expected to go much higher if the actual VLSI encoder chips are used.

VI. Conclusions

We have just shown the logic structure of a symbolic-slice VLSI RS encoder chip and the VLSI RS encoder system built by these chips. A design example has been given for a (255,223) VLSI RS encoder chip and VLSI RS encoder system. It has been shown that an RS encoder consists of four identical CMOS VLSI RS encoder chips connected together may replace around 40 CMOS IC's required by an encoder design optimized for discrete IC's. Besides the size advantage, the VLSI RS encoder also has the potential advantages of requiring less power and having a higher reliability.

References

[1] I.S. Reed and G. Solomon "Polynomial Codes Over Certain Finite Fields," J. Soc. Indust. Appl. Math., 8, pp. 300-304.

[2] G.D. Forney, Concatenated Codes, the MIT Press, Cambridge, Mass., 1966.

[3] J.P. Odenwalder, "Optimum Decoding of Convolutional Codes," Ph.D. dissertation, Syst. Sci. Dept., Univ. of Calif., Los Angeles, 1970.

[4] J.P. Odenwalder, et al. "Hybrid Coding System Study," submitted to NASA Ames Research Center by Linkabit Co., San Diego, Calif., Final Report, Contract No. NAS-2-6722, Sept. 1972.

[5] R.F. Rice, "Channel Coding and Data Compression System Considerations for Efficient Communication of Planetary Imaging Data," Technical Memorandum 33-695, Jet Propulsion Laboratory, Pasadena, CA., June 1974.

[6] R.F. Rice, "Potential End-to-End Imaging Information Rate Advantages of Various Alternative Communication Systems," JPL Publication 78-52, June 15, 1978.

[7] A. Hauptschein, "Practical, High Performance Concatenated Coded Spread Spectrum Channel for JTIDS," NTC '77, pp. 35:4-1 to 4-8.

[8] K.Y. Liu and K.T. Woo, "The Effects of Receiver Tracking Phase Error on the Performance of the Concatenated Reed-Solomon Viterbi Channel Coding System," NTC '80, pp. 51.5.1 to 51.5.5.

[9] W.W. Peterson and E.J. Weldon, Jr., Error Correcting Codes, The MIT Press, 1972.

[10] E.R. Berlekamp, "Better Reed-Solomon Encoders," presented at Calif. Inst. of Tech. EE Seminar, Pasadena, CA., Dec. 12, 1979.