A VLSI Single Chip 8-Bit Finite Field Multiplier

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

This article describes a VLSI architecture and layout for an 8-bit finite field multiplier. The algorithm used in this design was developed by Massey and Omura (Ref. 1). A normal basis representation of finite field elements is used to reduce the multiplication complexity. It is shown in this article that a drastic improvement has been achieved in this design. This multiplier will be used intensively in the implementation of an 8-bit Reed-Solomon decoder and in many other related projects.

I. Introduction

The era of VLSI digital signal processors has arrived and its impact is evident in many areas of research. The trend is to put more and more elements on a single silicon chip in order to enhance the performance and reliability of the system. These techniques are currently being used in the VLSI-based Reed-Solomon decoder using a small number of chips. Conventional decoders use several hundred chips. Our research has shown that it is possible to fabricate an entire decoder on a single chip.

Recently, finite field theory has found widespread applications. Examples include cryptography, coding theory and computer arithmetic. What is more, finite field arithmetic is the central part in the implementation of Reed-Solomon coders. Because there are quite a few finite field multiplications in the Reed-Solomon decoding algorithm, a small finite field multiplier is urgently needed for the implementation of a single VLSI chip Reed-Solomon decoder.

The major problem encountered in designing a small multiplier with the conventional method is the large quantity of hardware required. The conventional methods for implementing a finite field multiplier use either full parallel or table look-up algorithms. These methods lead to architectures that are not easily realized in a VLSI circuit. Massey and Omura (Ref. 1) recently developed a new multiplication algorithm for Galois fields based on a normal basis representation of field elements. In this article, a pipeline structure based on Massey and Omura's algorithm, developed in Ref. 2, is used to realize an 8-bit finite field multiplier. It is shown in this article that the chip area of this 8-bit multiplier is only about 1.3 times larger than the 4-bit multiplier designed and fabricated previously.

II. A Functional Description of the 8-Bit Multiplication Chip

The function f as described in Ref. 2 is chosen to be the following expression:

$$\begin{aligned} f(a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7; b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7) \\ = a_5b_0 + a_6b_0 + a_3b_1 + a_5b_1 + a_4b_2 + a_5b_2 + a_6b_2 + a_7b_2 \\ + a_1b_3 + a_4b_3 + a_2b_4 + a_3b_4 + a_0b_5 + a_1b_5 + a_2b_5 + a_6b_5 \\ + a_0b_6 + a_2b_6 + a_5b_6 + a_6b_6 + a_2b_7 \end{aligned} \quad (1)$$

where the generator polynomial of this finite field is

$$g(X) = X^8 + X^5 + X^3 + X + 1 \qquad (2)$$

There are a variety of different possible expressions of the function f; however, the one above was chosen because it has the least number (21) of terms (Ref. 3). Since each term in the expression represents a conduction line in the AND-plane of a PLA (programmable logic array), then obviously, the fewer the terms are, the less area will be spent in implementation.

An overall block diagram of the chip is shown in Fig. 1. In Fig. 1, Vdd and GND are power pins. The signals PHI-1 and PHI-2 are two non-overlapping phases of a system clock. The input bits of the multiplicand and multiplier are fed into the chip serially through the data input pins, DATA-IN1 and DATA-IN2. Similarly, the product of these two elements is transmitted out of the chip sequentially from the data output pin, DATA-OUT. The control signal LOAD is high for one bit time every 8 bits. The signal N-LOAD is just the logical complement of signal LOAD. Both the LOAD and N-LOAD signals are used for converting input data from serial type to parallel type.

Figure 2 shows the block diagram of an 8-bit finite field multiplier using Massey-Omura's normal basis algorithm. The circuit is divided into three units which are discussed in the following:

(1) Serial-to-Parallel Unit. This unit performs the serial to parallel conversion of input data sequence. In Fig. 3, R_i's and Q_i's, for $1 \leqslant i \leqslant 2$, represent 7-bit and 8-bit shift registers with parallel load, respectively. The input bits of the multiplicand and multiplier are first stored in the R_i registers. These data will then be loaded into the Q_i registers for every eight clock cycles. To reiterate, when signal LOAD is low and N-LOAD is high, data come in bit by bit. At the eighth clock cycle, signal LOAD is high and signal N-LOAD becomes low. Data in the R_i registers will then be loaded into Q_i registers. The output of Q_i registers is the input of the AND-generation unit.

(2) AND-Generation Unit. This unit generates the ANDed terms of the input bits of the multiplicand and multiplier. The AND functions are configured in a structure of AND-plane of PLA because of the regularity and modularity that a PLA possesses. The inputs to this unit are the complemented values of the input bits of the multiplicand and multiplier. The output of this unit will be the ANDed terms of the input. These ANDed terms will be sent to the exclusive-or unit.

(3) Exclusive-Or Unit. In finite field arithmetic, if the ground field is $GF(2)$, addition is just the exclusive-or operation. Since there are twenty additions in Eq. (1), twenty exclusive-or cells are required for the conventional method to implement these exclusive-or operations. Because each exclusive-or cell needs a substantial amount of chip area, 20 of these will consume so much area such that a small multiplier will not be possible.Consequently, this will prohibit the implementation of a single chip 8-bit Reed-Solomon decoder. In our design, an alternate way for implementing the exclusive-or operation is used. This new technique was developed and used in the fabrication of the Multicode Convolutional Encoder chip (Ref. 5).

Figure 4 shows the block diagram of the conceptual exclusive-or array. This array comprises twenty-one subcells. Each subcell performs a switching operation. As shown in Fig. 5, if input A is a "one," i.e., it is high, then its complement signal $\overline{A}$ is low and signals $S1$ and $S2$ will be switched. On the other hand, if signal $\overline{A}$ is low, then A is high and signals $S1$ and $S2$ will not be switched at this moment.

There are two signal paths through each row of the subcell. At the extreme left of the array, a "one" is connected to one of these and a "zero" to the other. Each time this pair of signals passes through a subcell, they exchange places if the corresponding output of AND-generation unit is "one." In this way, the pair of signals has gone through a number of path exchanges equal to the number of ones in the AND-generation unit output. If the number of ones of the AND-generation unit output is even, then the signals are in the same places as they started. If the number of ones is odd, they will come out reversed.

III. The Estimated Performance of the 8-Bit Multiplier

The 8-bit multiplier chip was designed using the UNIX-based integrated CAD system (Ref. 6). The entire chip was simulated on a general purpose computer using ESIM (a logic level simulation program; Ref. 7) and SPICE (a transistor level simulation program; Ref. 8). The layout of the multiplier was accomplished using the program CAESAR (Ref. 9). LYRA (Ref. 10) was used to check the resulting layout against a set of geometric rules supplied by the fabricator. Timing simulation was done using CRYSTAL (Ref. 11). The circuit comprises about 2000 transistors.

The chip described here was sent out for fabrication through the MOSIS service (Ref. 12). The technology used was 3 μm NMOS. When the completed chips return they will be evaluated and tested. The layout of the multiplier is shown in Fig. 6. The area of this chip is $1200 \times 900\ \mu m^2$ while the chip area of a 4-bit finite field multiplier designed previously is $1000 \times 800\ \mu m^2$. The 8-bit one is only 1.3 times larger than the 4-bit. The estimated operation speed of this chip is 10 MHz and the estimated power consumption at this frequency is 30 mW.

References

1. J.L. Massey and J.K. Omura, U.S. Patent Application of "Computational Method and Apparatus for Finite Field Arithmetic," submitted in 1981.

2. C.C. Wang, T.K. Truong, H.M. Shao, L.J. Deutsch, J.K. Omura and I.S. Reed, "VLSI Architecture of Computing Multiplications and Inverse in $GF(2^m)$," *TDA Progress Report 42-75*, pp. 52–64, 1983. Jet Propulsion Laboratory, Pasadena, California.

3. C.C. Wang, "Computer Simulation of Finite Field Multiplications Based on Massey-Omura's Normal Basis Representation of Field Elements," private communication, 1985.

4. C.M. Mead and L. Conway, *Introduction to VLSI Systems*, New York, Addison-Wesley Publishing Company, 1980.

5. L.J. Deutsch, "A VLSI Implementation of a Multicode Convolutional Encoder," *TDA Progress Report 42-72*, pp. 61–69, 1985. Jet Propulsion Laboratory, Pasadena, CA.

6. L.J. Deutsch, "An UNIX-Based CAD System for the Design and Testing of Custom VLSI Chips," *TDA Progress Report 42-81*, pp. 51–62, 1985. Jet Propuslion Laboratory, Pasadena, CA.

7. C. Terman, "ESIM – An Event Driven Simulator," Technical Memorandum, Electrical Engineering Department, Massachussette Institute of Technology, 1977.

8. L.W. Negal and D.O. Pederson, "SPICE – Simulation Program with Integrated Circuit Emphasis," *Memorandum No. ERL-M382*, Electronics Research Laboratory, University of California, Berkeley.

9. J. Ousterhout, "CAESAR – Editing VLSI Circuits with CAESAR," Computer Science Division, Technical Memorandum, Electrical Engineering and Computer Science Department, University of California, Berkeley, April 21, 1982.

10. J. Ousterhout, "LYRA – A Design Rule Checker," Computer Science Division, Technical Memorandum, Electrical Engineering and Computer Science Department, University of California, Berkeley, April 21, 1982.

11. J. Ousterhout, "Using Crystal for Timing Analysis," Computer Science Division, Technical Memorandum, Electrical Engineering and Computer Science Department, University of California, Berkeley, March, 1983.

12. The MOSIS Project, The MOSIS System (What It Is and How to Use It). Information Science Institute, Univeristy of Southern California, *Publication ISI/TM-84-128*, Marina Del Rey, CA.

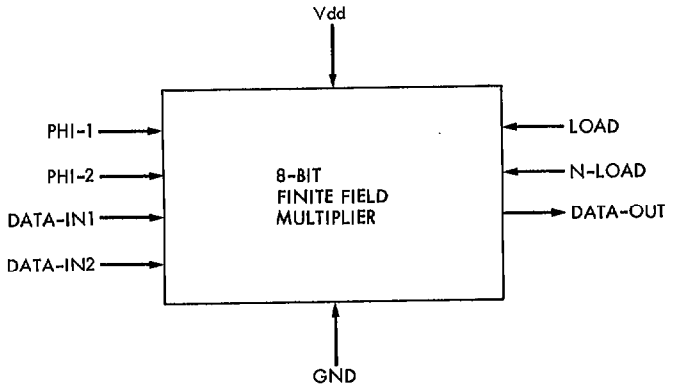

Fig. 1. The symbolic diagram of an 8-bit finite field multiplier

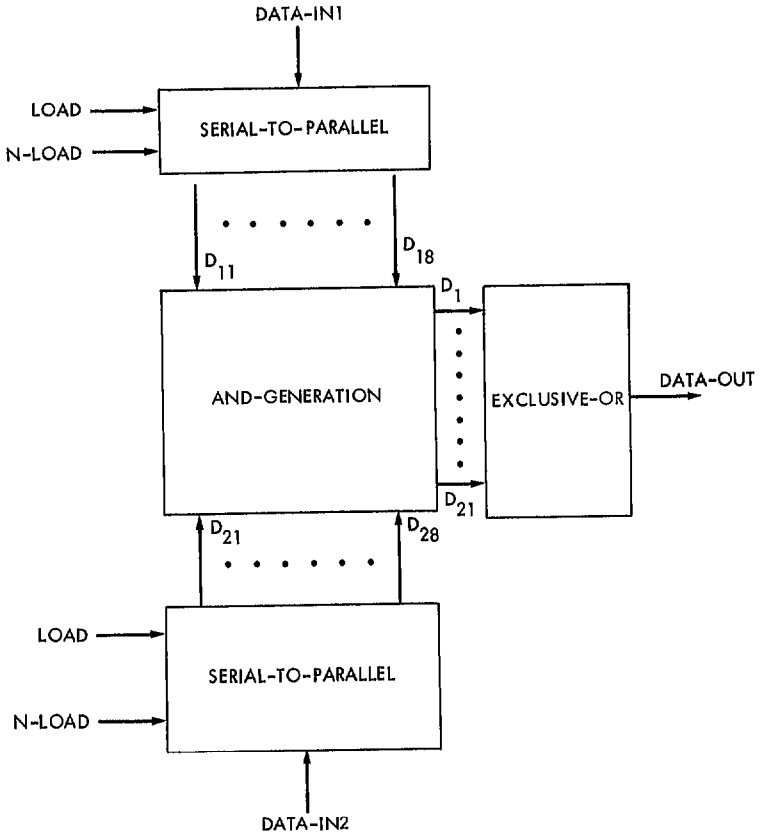

Fig. 2. The block diagram of an 8-bit finite field multiplier

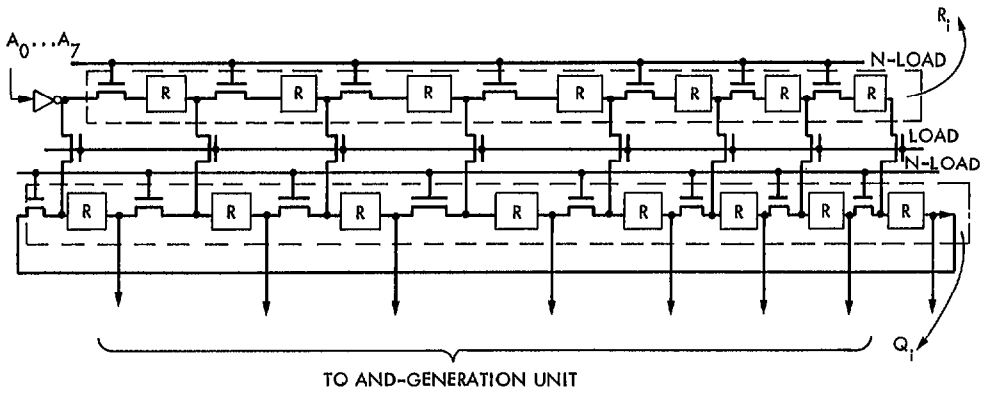

Fig. 3. The logic diagram of R_i's and Q_i's registers

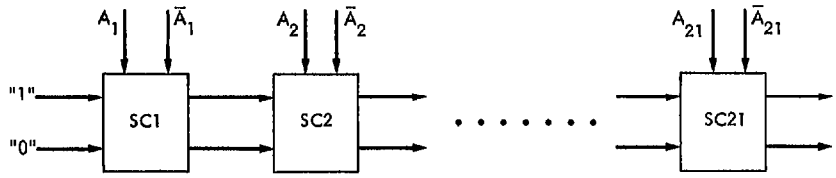

Fig. 4. The block diagram of conceptual exclusive-or array

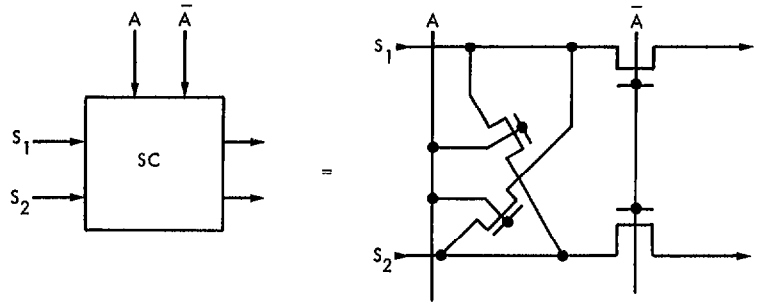

Fig. 5. The circuit diagram of a subcell in the conceptual exclusive-or arrays

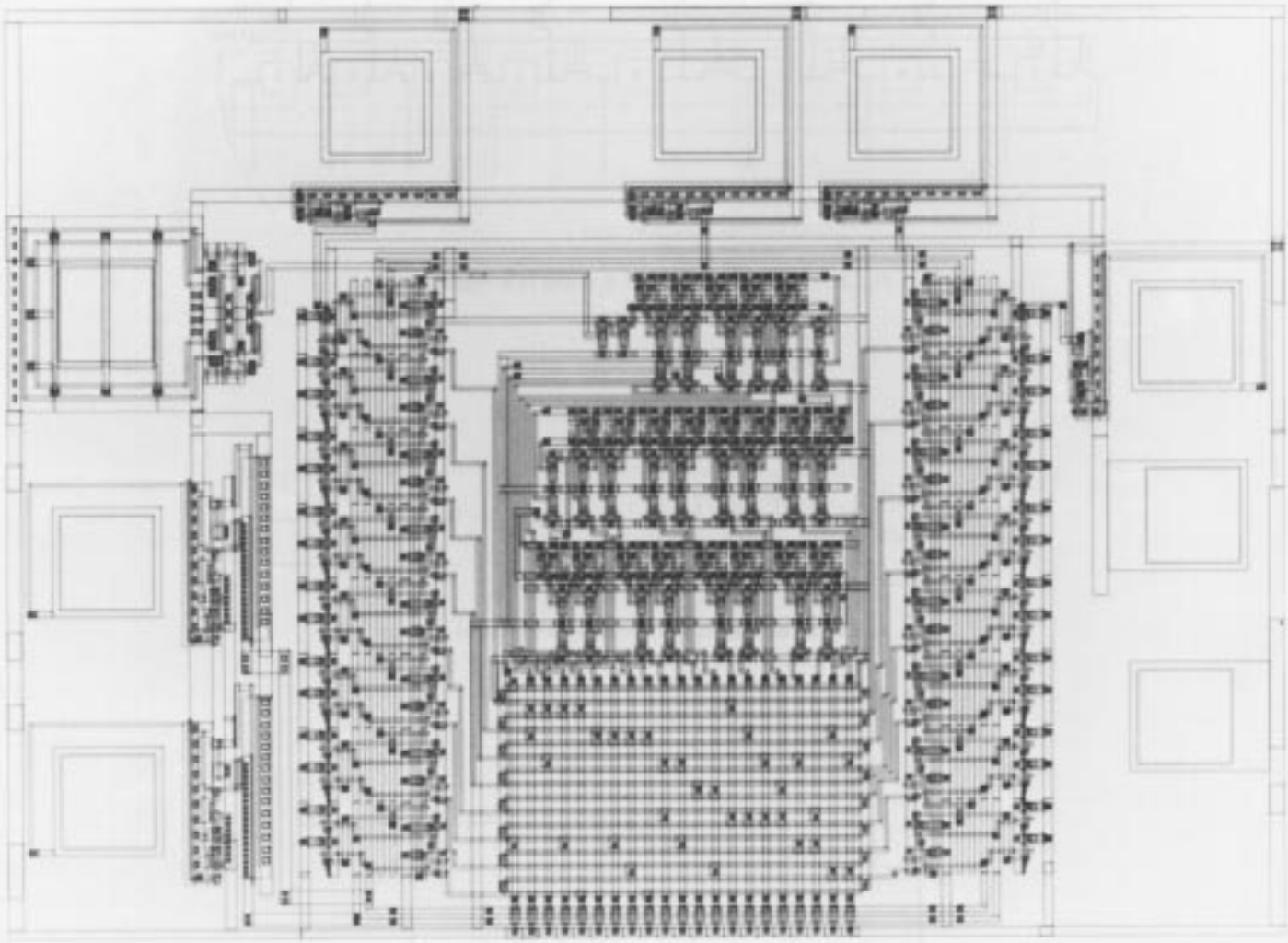

Fig. 6. The layout of an 8-bit finite field multiplier