Preprint 2316 (A4)

A LSI for Reed-Solomon Encoder/Decoder

Ken Onishi*, Kazuhiro Sugiyama*, and Yoshinobu Ishida*
Yoshitsugu Kusunoki**, and Tetsuya Yamaguchi**

* Consumer Electronics Development Laboratory
** Communication Equipment Works
Mitsubishi Electric Corporation

**Presented at
the 80th Convention
1986 March 4-7
Montreux, Switzerland**

AUDIO ES

AES

This preprint has been reproduced from the author's advance manuscript, without editing, corrections or consideration by the Review Board. The AES takes no responsibility for the contents.

Additional preprints may be obtained by sending request and remittance to the Audio Engineering Society, 60 East 42nd Street, New York, New York 10165 USA.

AN AUDIO ENGINEERING SOCIETY PREPRINT

AN LSI FOR REED-SOLOMON ENCODER/DECODER

Ken Onishi*, Kazuhiro Sugiyama*, and Yoshinobu Ishida*
Yoshitsugu Kusunoki**, and Tetsuya Yamaguchi**

*Consumer Electronics Development Laboratory
**Communication Equipment Works
Mitsubishi Electric Corporation

ABSTRACT

The Reed-Solomon Code is widely used for an error correcting code in digital audio. The authors have developed an LSI for the Reed Solomon Encoder/Decoder. This LSI has a characteristic that the code length, the minimum code distance, the performance of the decoder can be variable by changing micro-program.

1. INTRODUCTION

The Reed-Solomon Code (1) is a generally used error correcting code, which has been incorporated with a variety of practical equipment, i.e., the professional use digital audio tape recorder (2), compact disc player, satellite communication apparatus, etc., having a prospective utilization for consumer's DAT (Digital Audio Tape Recorder) (3), optical disk apparatus, etc.

The Reed-Solomon Code is one of the maximum distance separable codes, so that it is favoured with high efficiency, and the one which occupies the main stream currently used is that over galois field GF (2^8). This is because of the fact that the code construction is favoured with high flexibility since the code length and the code distance are available up to 255 over GF(2^8), and it is easy to realize in hardware thanks to substantial progress of LSI technology in recent years.

Currently the authors have succeeded in the development of a LSI facilitating the encoding and decoding of these Reed-Solomon Codes, so that, in this proper, descriptions are to be given for the outline of this LSI, hardware and software architectures thereof, software development supporting system, and several examples of their applications.

2. OUTLINE OF THE LSI

This LSI executes encoding and decoding of Reed-Solomon Code by a means of incorporating program control and facilitating real time processing when it is used in the field of digital audio (4) (5). In spite of this LSI having plentiful kinds of commands, its hardware is very simple and of low price, favoured with various outstanding features, the main of which are given below:

1. The code length and the minimum code distance facilitate flexible arrangement of Reed-Solomon Codes over GF (2^8) within a range from 2 through 255; since the code length and the minimum code distance may be specified by means of programming, resulting in high flexibility.

2. It can be in response to various kinds of encoding and decoding algorithm; there are Peterson algorithm (6), Berlekamp algorithm (7), and Euclideon algorithm, (8), for decoding methods, whichever one of the above being able to be used in this LSI.

3. The time sharing processing of multiple numbers of encoders and decoders are available by means of a single LSI. A high speed performance is available, and multiple numbers of programs can be stored, so that the cost down of the equipment can be achieved by effective time sharing processing.

This LSI is so constructed that, as shown in Fig. 1, it may function with three parts configuration, together with the External Program ROM, and External Table ROM, the former being written-in with a microprogram making the LSI function as encoder/decoder, and the latter is written-in with a unit conversion table over galois field which is required en route of calculation. The outstanding functions of this LSI are given below:

1. Calculating Function
   Calculations are to be undertaken over galois filed GF (2^8), taking $X^8+X^4+X^3+X^2+1$ as a primitive polynomial, in which one step of complex calculation combining multiplications and additions is available, ALU (Arithmetic Logic Unit), in which additions and subtractions are conducted, being also provided.

2. Various Kinds of Programming Functions
   In the following, specifications in relevance to programming are to be shown:

Item	Content
Program step number	1024 steps in the maximum
1 instruction	48 bits
Number of programs	8 programs in the maximum
Subroutine	1 fold

Jump	Indirect/direct address Unconditional, conditional
Flag	2 for each Input/output

3. Memory and Input/Output
   2 series of RAM (64 bytes for each) are contained internally, symultaneous accessing being available; input/output terminal being provided with parallel parts, the transferring by programming and DMA (Direct Memory Access) for high speed transfer are available.

3. HARDWARE ARCHITECTURE

This Reed-Solomon encoder/decoder LSI incorporates the configurations as given below in order to realize the high speed calculation processing in the error correcting signal processing.

- Complex calculation GLU
- 48 bit microprogram
- Table ROM
- Dual internal RAM
- Dual parallel I/O interface
- Register configuration suited for calculation processing

The main function block of LSI is composed of GLU, memory, I/O, α-gen, ALU, etc., the configuration of which is given in Fig. 2.

3.1 GLU

The analysis of encoding/decoding algorithm results in the fact that realization of a single step of complex calculation can clearly show substantial enhancement of processing speed, so that the LSI incorporates GLU configuration as shown in Fig. 3 so as to facilitate the complex calculations as shown below:

1. $(X * Y) + Z$
2. $X + Z$
3. $X * Y$
4. $(X + Z) * Y$

$*$: multiplication over GF (2^8)
$+$: mod 2 addition

The input is provided with registers X, Y, and Z, the output being provided with registers respectively as U.V. and W, so as to facilitate pipeline processing, and further function to store the above said calculation results again into the input resistor is provided, letting them be utilized as the source data for the next command.
Since the multiplier shown in Fig. 3 requires high speed processing, so that a parallel multiplier is incorporated, the circuit shown on Fig. 4 is the one for the parallel multiplier,

in the following a simple description of how this circuit can be deduced shall be given.
Assuming the input $\mathfrak{A}=(a_7, a_6, a_5, \ldots a_0)$ and $\mathbb{b}=(b_7, b_6, b_5, \ldots, b_0)$ and the output $\mathbb{C}=(c_7, c_6, c_5, \ldots c_0)$; then they can be expressed as:

$$\begin{aligned}\mathfrak{A}&=a_7\alpha^7+a_6\alpha^6+a_5\alpha^5+\ \ldots\ldots\ +a_0\\ \mathbb{b}&=b_7\alpha^7+b_6\alpha^6+b_5\alpha^5+\ \ldots\ldots\ +b_0\\ \mathbb{C}&=c_7\alpha^7+c_6\alpha^6+c_6\alpha^5+\ \ldots\ldots\ +c_0\end{aligned} \qquad \ldots\ldots\ (1)$$

The multiplication gives:

$$\begin{aligned}\mathfrak{A}\cdot\mathbb{b}&=(a_7\alpha^7+a_6\alpha^6+a_5\alpha^5+\ \ldots\ +a_0)(b_7\alpha^7+b_6\alpha^6+b_5\alpha^5\ \ldots\ +b_0)\\ &=c_7\alpha^7+c_6\alpha^6c_5\alpha^5+\ \ldots\ +c_0\\ &=\mathbb{C}\end{aligned} \qquad \ldots\ldots\ (2)$$

where,

$$\alpha^8=\alpha^4+\alpha^3+\alpha^2+1 \text{ (where } \alpha \text{ is the root of the primitive polynomial.)} \qquad \ldots\ldots\ (3)$$

Execute the calculation of equation (2), utilizing the above given formula: the result will be:

$$\mathbb{C}=f(\mathfrak{A},\mathbb{b}) \qquad \ldots\ldots\ (4)$$

Fig. 4 shows the above relation represented in a circuit diagram, in which a gate delay 8 stage high speed calculation is realized.

3.2 Memory

The memory of this LSI is composed of 3 kinds, namely:

- Instruction-ROM (External)
  1024 words (max) x 48 bits (24 bits x 2 reading)
- Table-ROM (External)
  1024 words (max) x 8 bits
- Internal-RAM
  128 (64 x 2) words x 8 bits

Instruction-ROM:
These microprograms, such as the encoder, decoder, etc. shall be written in the Instruction ROM; the microinstruction is composed of one word 48 bits, however, in this LSI, the instruction bits are divided into 24 bits x 2 sections, so as to transfer one word by two times, (number of program ROM being to be reduced); a microprogram having 1024 steps in maximum is available, and besides function of subroutine stack of one level is also provided, so that effective size of programming is virtually enlarged; there is a Mode-set function provided, in which selections of 8 kinds of starting addresses are feasible in case of the program start.

Table-ROM:
The Table-ROM is to be used for unit exchange of the data, e.g., letting an error location be i,

1. $\alpha^{i} \to \alpha^{-i}$ (unit inversed)
2. $i \to \alpha^{i}$ ($i \to \alpha^{-i}$)
3. $\alpha^{i} \to i$ ($\alpha^{i} \to -i$)
4. 2 error location table

it is to be used for the memory of the fixed data as given above.

Internal-RAM:
The internal-RAM is used for the memory of variable such as temporary data en route of real time processing, syndrome data, error location, etc. There are two independent systems prepared in order to enhance the effectiveness of the calculation processing, (i.e., IRAM-A and IRAM-B).
RAM can be defined whichever source or destination register, the address of which is specified by the counter for data points (CA, CB, CC) or by direct address from Instruction-ROM.

3.3 THE OTHER FUNCTIONS

Data Processing:
A Dual-Data-Bus system (A-Bus and B-Bus) is employed so as to execute the complex calculating function and independent 2 system Internal-RAM effectively; all of the blocks such as the Instruction-ROM, Internal-RAM, GLU, ALU, I/O Interface, etc., are connected with the Bus, so that various kinds of processings may be conducted appropriately.

ALU:
In which addition and subtraction calculus is available to be used for the degree of polynomial mainly.

α-generator:
To generate the root α of primitive polynomial $X^8+X^4+X^3+X^2+1=0$ step by step, being to function effectively in case of syndrome calculation.

Control Flag:
There are I/O flag, (FI1, FI2, FØ1, FØ2), counter flag, (F3 and F4), calculation flags of GLU and ALU, (F5, and F6), etc., are available to be used for input/output control or conditional jump, etc.

I/O Interface:
There are the interfaces available as shown below:

- 1st-port	8 bit	Output
- 2nd-port	8 bit	Input/output
- 3rd-port	2 bit	Input/output

The 1st port is connected with CB or B-Bus, mainly to function as a data pointer in response to the external data bank; the 2nd port is an input/output port, which is connected with X-Register in case of input, and with A-Bus in case of output, so as to be used mainly for data input/output; the 3rd-port is an input/output port of 2 bits to be used for input/output of control flag. There are two methods of data transfer available, namely the method by instruction, and the one by DMA, being selected whichever Program/DMA by program.

4. SOFTWARE ARCHITECTURE

The Reed-Solomon encoder/decoder LSI, as shown in Fig. 5, incorporates 48 bit construction of horizontal type microcommand which is suitable for signal processing; among the 48 bits, 41 bits are incorporated for the control program for each individual block, and at the same time to be constructed so as to undertake parallel processings; the microcommand is encoded in 7 bits to be prepared in 14 kinds. First, the control program for each block can be categorized as follows:

1. Data transfer between Internal-RAM and Data-Bus, and indirect address counter control.
   There are two kinds of adressing modes available:

   - Direct Addressing: To be specified by Instruction ROM.
   - Indirect Addressing: To take the outputs of counters (CA, CB and CC) having functions of up, down, and load as adresses.

2. Conditional Jump Instruction
   There are conditional instructions available as shown below:

   - Input/output control signal FI1, FI2
   - To indicate the status of indirect address F3 and F4
   - To indicate the calculation results of GLU and ALU F5 and F6

3. Calculation Instruction

4. Controls of Various Registers and Selectors

   Besides the above, there are instructions available as given below mainly as the microinstructions.

   1. Subroutine call, and return
   2. Arithmetic calculation (addition and subtraction)
   3. Flag control

4. DMA control
5. Table ROM select

The instruction command list is given on Table 1.

5. PROGRAM DEVELOPMENT SYSTEM

There are two kinds of tools employed for the development of Reed-Solomon Encoder/Decoder LSI, one of which is a tool used for development of microprogram, and the other one is a hardware test tool in which this LSI is incorporated; the comprehensive configuration diagram of this system is given in Fig. 6.

5.1 DEVELOPMENT OF PROGRAM

The program has been created on the main computer, the program produced being written in the Program-ROM by means of ROM writer. Fig. 7 gives the Block diagram of the supporting software; in the following descriptions are to be given, referring to the below:

1. Editor
   The editor is responsible for executing the creation and edition of the microprogram, incorporating various commands, such as EDIT, DELEAT, INSERT, LIST, SAVE, LOAD, NEW, etc., which the source may be filed in a floppy disk.

2. Assembler
   The function of the assembler is to convert the program into machine language for LSI control.

3. Tracer
   This is the most outstanding tool among the tools incorporated in the development, which can trace step by step the data on the hardware, such as Register, IRAM, Counter, GLU, ALU, Flag, I/O, etc., by the function of which debugging the program is available without difficulty, contributing to the reduction of development term of software. Fig. 8 shows an example of tracing.

4. Encoding/Decoding Simulator
   Associated with development of a microprogram for encoder and decoder, the simulation of encoder/decoder on the calculation formula is required to check if the results of tracing is correct or not. For the decoding algorithm, the error location polynomial is to be obtained by Euclidean relation from syndrome, so that the Euclidean algorithm is to be employed (8).
   Fig. 9 shows the flow of parity check between the calculation results of the hardware tracer and the decoding simulation results by means of calculating equation. The checking capability for those such as syndrome, key

equation, error location, error value, decoding results, are provided for the checking function.
Incorporating random numbers for input data, the program can be made up to be the complete one by means of repeating the simulation of the check flow several times.

5.2 HARDWARE TEST

This is to execute what has been simulated by decoder check by software given in Fig. 9 exclusively, by means of real hardware, referring to the system shown in Fig. 7. To the LSI the encoded data accompanied with error from the data generator are to be inputted; where the LSI functions as a decoder, the result of calculation being taken-in by a data analizer to be checked if the results of decoding is correct or not on the computer. In this system a complete decoder check, including I/O interface, can be conducted; of course, the checking function is also available, even if the LSI is used as an encoder.

6. APPLICATION

It has been revealed that this Reed-Solomon encoder/decoder LSI is able to be in response to various kinds of Encoder/Decoder algorithms, and in this section, as examples, the cases where it is applied as an encoder, and where it is also applied to a syndrome calculator shall be introduced.

6.1 ENCODER

Assuming the case where the (16, 12, 5) Reed-Solomon Code is used for encoding, the generator polynomial of the minimum code distance d=5 shall be given by the following equation:

$$g(Z) = \pi_{i=0}^{3} (Z + \alpha^{i})$$

$$= (Z + \alpha^{0})(Z + \alpha^{1})(Z + \alpha^{2})(Z + \alpha^{3})$$
$$= Z^{4} + \alpha^{75}Z^{3} + \alpha^{249}Z^{2} + \alpha^{78}Z + \alpha^{6}$$

where an encoder (6), incorporating a subtraction circuit of polynomial, shall be taken into consideration, utilizing the characteristics in that the code language can be exactly divisible by the generater polynomial. If it is taken as a special purpose hardware, the block diagram shall be that given in Fig. 10, (a). The number of calculation steps required for this calculation shall be as given below, when it's calculated roughly:

EXOR	4 x 12 times
Multiplier	4 x 12 times
Data input	12 times
Data output	4 times
Total	112 times

Fig. 11 gives the program in which this encoding algorithm is realized by means of the Reed-Solomon encoder/decoder LSI. Generally when the calculation is to be executed by means of a processor, it is required to spend a considerable time for transferring between the Register, Internal-RAM, GLU, etc., besides the above said calculation. However, in case of the program given in Fig. 12, this encoding is realized by program size of 15, and an execution step number of 64, which is embodied by effective utilization of complex calculation of (X * Y) + Z, Dual Internal-RAM, and Dual-Data Bus.

6.2 SYNDROME GENERATOR

Let us consider of the syndrome generator of the (16, 12, 5) Reed-Solomon Code as like as the case of 6.1, the calculating equation of syndrome may be given by the following: Assuming the receiving signal $\mathbb{r}=(r_0, r_1, r_2, \ldots, r_{15})$ and Syndrome $\mathbb{S}$ $=(s_0, s_1, s_2, s_3)$,

$$
\begin{aligned}
r_0 + r_1 \quad + r_2 \quad + r_3 \quad + \ldots + r_{15} \quad &= s_0 \\
r_0 + r_1\alpha \quad + r_2\alpha^2 + r_3\alpha^3 + \ldots + r_{15}\alpha^{15} &= s_1 \\
r_0 + r_1\alpha^2 + r_2\alpha^4 + r_2\alpha^6 + \ldots + r_{15}\alpha^{30} &= s_2 \\
r_0 + r_1\alpha^3 + r_2 a6 + r_3\alpha^9 + \ldots + r_{15}\alpha^{45} &= S_3
\end{aligned}
$$

from the above equation, they can be realized by hardware simply as given in Fig. 10 (b), where the numbers of calculations required are:

EXOR	15 x 4 times
Multiplier	15 x 3 times
Data input	16 times
Total	121 times

Fig. 12 gives the result which is realized by the Reed-Solomon Encoder/Decoder LSI as for the Syndrome Generator, where the program size is 18, steps being 70 in the number of execution steps for the purpose of the realization.

7. CONCLUSIONS

An LSI, in which the encoding/decoding of the Reed-Solomon Code over GF (2^8) are made available has been developed; the main outstanding feature of this LSI is as follows:

1. It can be ready to correspond a variety of code lengths, and code distances, since its performance is being controlled by a programable ROM.
2. The development of the program is conducted effectively by means of a development supporting system.
3. In spitè of high speed operations, this LSI is favoured which low consumption of electric power and low price, thanks to CMOS construction being incorporated.

This LSI is mounted on a digital audio recorder (9) incorporating the 2-channel PD Format, and this sort of LSI is exchangeable as in the like manner as μ-Processor, or DSP (Digital Signal Processor), so that a prospect of considerable prevalence in the market may be foreseen in the years to come.

8. ACKNOWLEDGEMENT

Authors are very much grateful to the gentlemen in Mitsubishi Electric Co. who have been supporting this project, associated with development of this LSI, namely Mr. M. Itoga, Mr. S. Kunii, and Mr. K. Ishii, and also to our staff who have been participating in this project, namely Mr. M. Hamamura, Mr. T. Izaki, and others, and Mr. B. Bearman for his useful advices.

REFERENCE

(1) I.S. Reed & G. Solomon, "Polynominal Codes over Certain Finite Field, "J. Soc. Indust. Appl. Math., 8, pp.300 ∿ 304, 1960.

(2) K. Tanaka, T. Yamaguchi and Y. Sugiyama; "IMPROVED TWO CHANNEL PCM TAPE RECORDER FOR PROFESSIONAL USE", AES 64th Conv. No. 1533 (G-3), Nov. 1979

(3) Y. Ishida, M. Ishida, K. Goto, K. Endo, Y. Osuga, K. Ido; "ON THE DEVELOPMENT OF A ROTARY-HEAD DIGITAL AUDIO TAPE RECORDER", AES 79th Conv. No. 2271 (A-5), Oct. 1985

(4) K. Sugiyama, K. Onishi, S. Kunii; "ENCODER AND DECODER FOR REED SOLOMON CODES", Papers of Technical Group on Electric Audio of I.E.C.E. of Japan, 84-61 (1984)

(5) K. Sugiyama, K. Onishi, S. Kunii; "RS FUGOU SIGNAL PROCESSOR NO KAIHATSU SHIEN SYSTEM", Deñki Kankei Gakkai Kansai Shibu Rengou of Japan, G15-11, 1984

(6) W. Wesley Peterson and E. J. Weldon, Jr.; "Error correcting Codes", MIT Press, 1961

(7) E.R. Berlekamp: "Algebraic Coding Theory", Mc Graw-Hill, 1968

(8) Y. Sugiyama et al., "An Erasures and Errors Decoding Algorithm for Goppa Codes", IEEE Tranc, IT, Mar. 1976

(9) Y. Ishida, K. Onishi, K. Sugiyama, T. Yamaguchi, Y. Kusunoki; "A PROFESSIONAL USE 2-CHANNEL DIGITAL AUDIO RECORDER ADOPTING IMPROVED SIGNAL FORMAT", AES 80th Conv. Mar. 1986

TABLE 1 INSTRUCTION TABLE

Type	Operation
IRAM - A	Direct Read Indirect Read Direct Write Indirect Write
IRAM - B	Direct Read Indirect Read Direct Write Indirect Write
Counter-A	Nop Up Down Load
Counter-B	Nop Up Down Load
Counter-C	Conditional Up
Jump	Nop Conditional (FI 1, FI 2, F3, F4, F5, F6) Unconditional
GLU	$(X \ast Y) \oplus Z$ $X \oplus Z$ $X \ast Y$ $(X \oplus Z) \ast Y$
Micro	STP DMA SBR ALF LAT ADD SUB FSH EXR FCM EFM CFM DM OTS

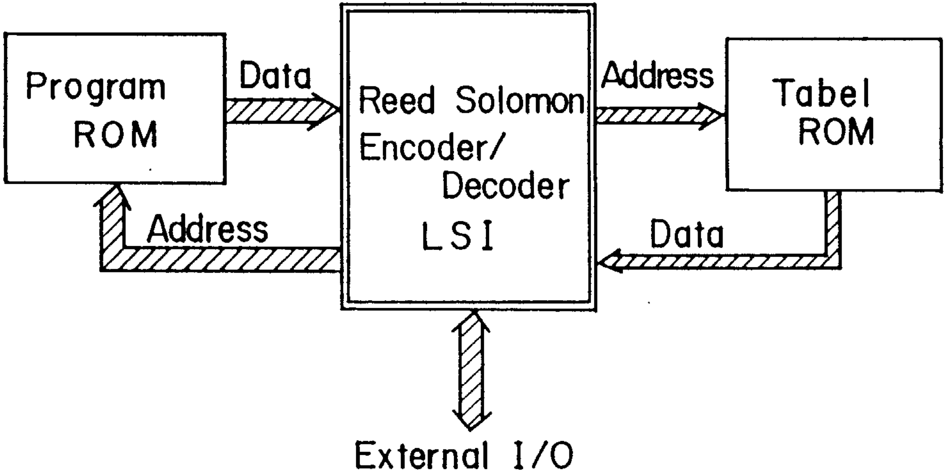


FIG.1 SYSTEM BLOCK DIAGRAM

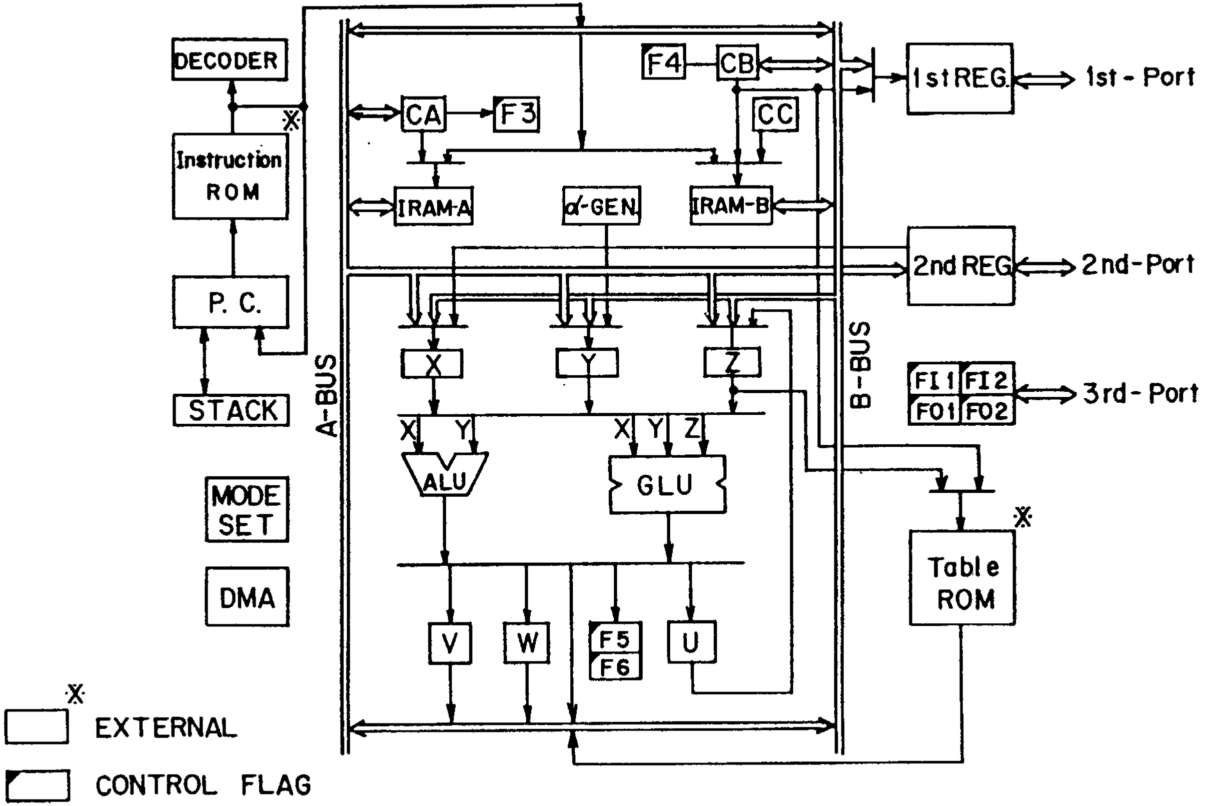


FIG.2 BLOCK DIAGRAM OF THE LSI

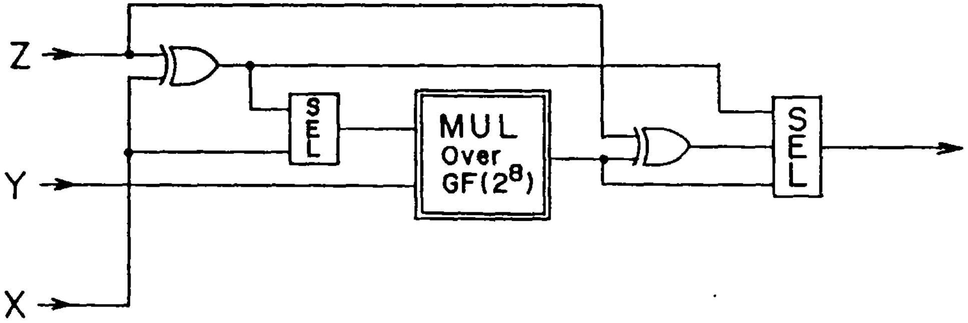


FIG.3 BLOCK DIAGRAM OF GLU

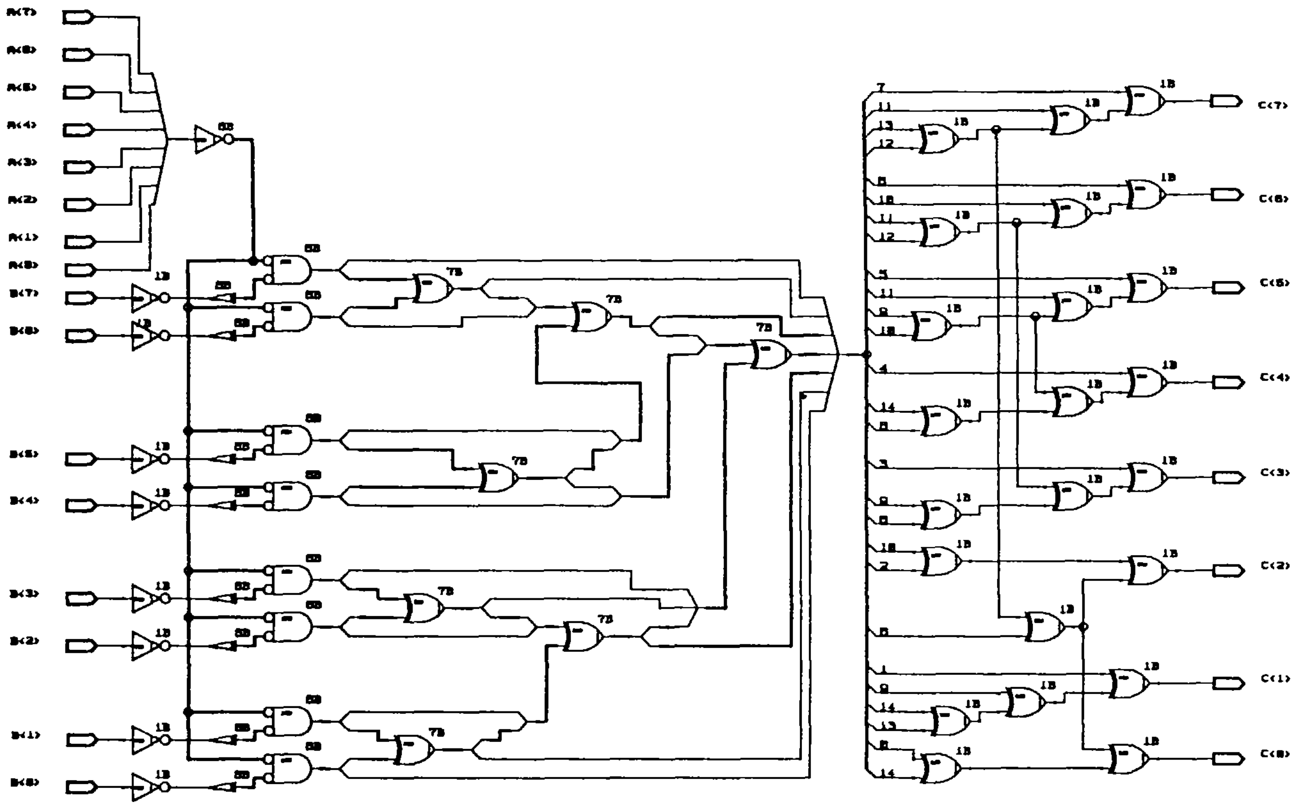


FIG.4 PARALLEL MULTIPLIER OVER GF(2^8)

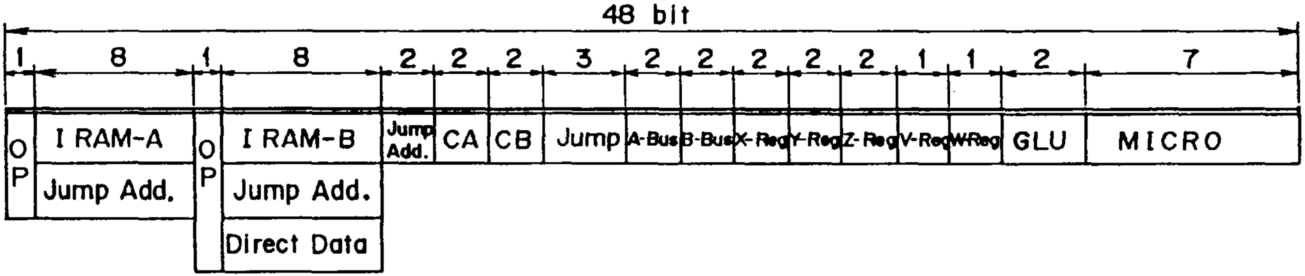


FIG. 5 INSTRUCTION FORMAT

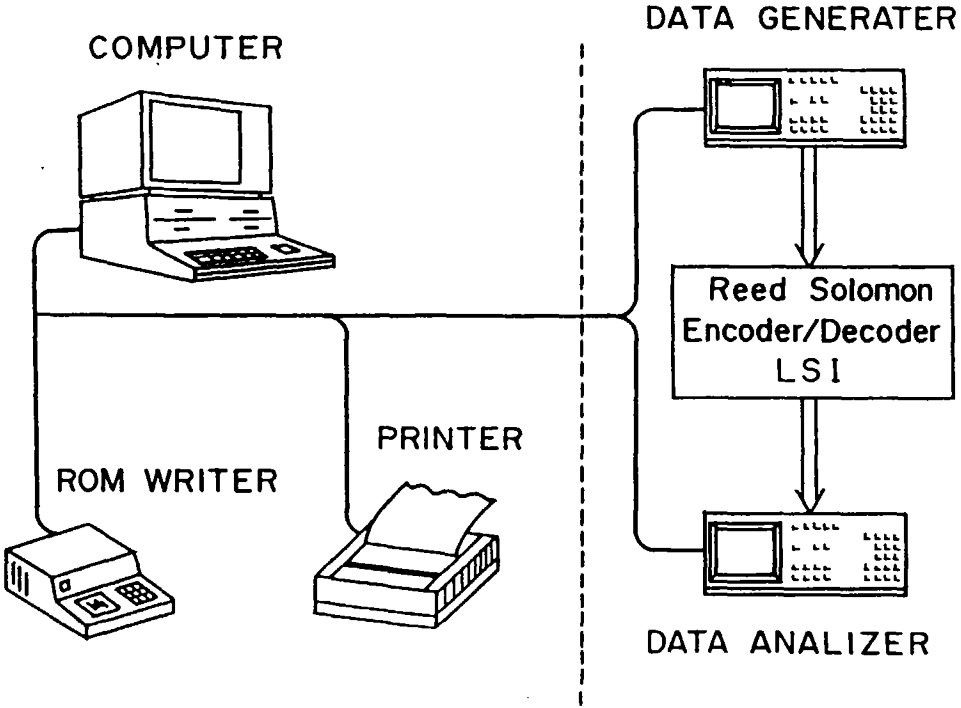


FIG. 6 PROGRAM DEVELOPMENT SYSTEM

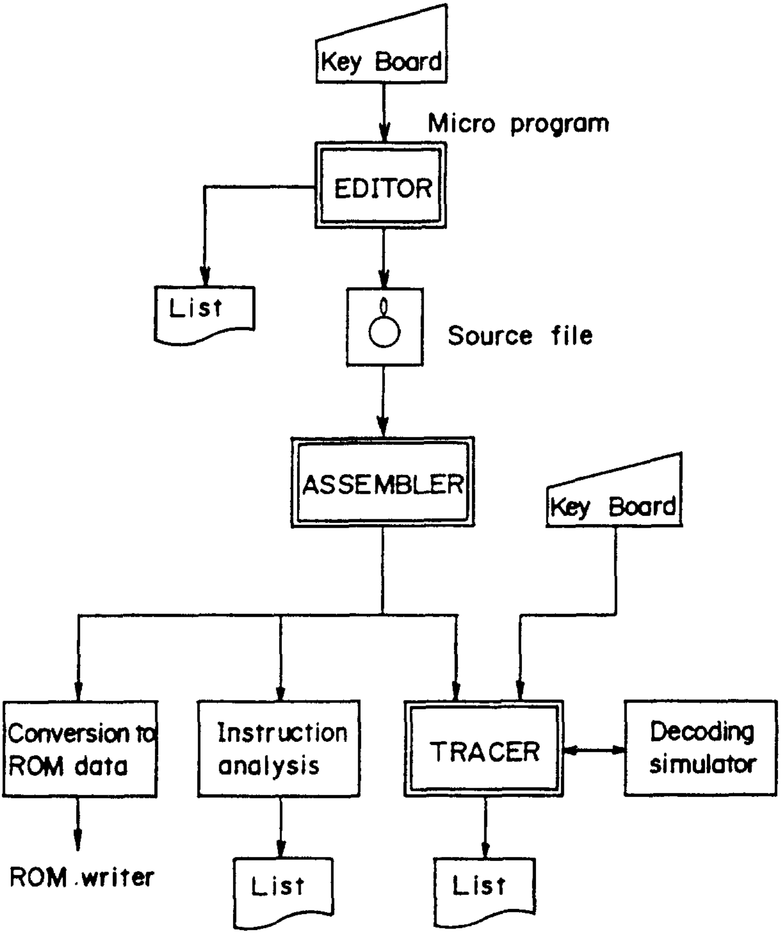


FIG.7 SOFTWARE FOR PROGRAM DEVELOPMENT

```
TRACE  LINE   CA  CB  X   Y   Z   GLU  V   W   U   IN  OUT  F123456  A  B
0000   00005  00  0B  --  --  --  --   --  --  --  --  --   -------  -  -
0001   00010  --  --  00  --  00  --   --  --  --  01  --   0------  -  -
0002   00020  --  0A  01  --  --  00   00  00  00  --  --   -------  -  -
0003   00030  --  --  01  0F  00  01   --  --  --  --  --   -------  -  -
0004   00040  --  --  --  36  00  0F   0F  --  --  01  --   0------  -  -
0005   00050  --  --  --  78  00  36   --  --  36  --  --   -------  -  -
0006   00060  --  --  --  40  --  78   78  --  --  --  --   ---1--   -  -
0007   00070  --  09  01  --  0F  40   --  40  --  --  --   -------  -  -
0008   00080  --  --  0E  0F  36  0E   --  --  --  --  --   -------  -  -
0009   00040  --  --  --  36  78  6C   6C  --  --  01  --   0------  -  -
0010   00050  --  --  --  78  40  61   --  --  61  --  --   -------  -  -
0011   00060  --  --  --  40  --  AA   AA  --  --  --  --   ---1--   -  -
0012   00070  --  08  01  --  6C  A7   --  A7  --  --  --   -------  -  -
0013   00080  --  --  6D  0F  61  6D   --  --  --  --  --   -------  -  -
0014   00040  --  --  --  36  AA  30   30  --  --  01  --   0------  -  -
0015   00050  --  --  --  78  A7  66   --  --  66  --  --   -------  -  -
0016   00060  --  --  --  40  --  15   15  --  --  --  --   ---1--   -  -
0017   00070  --  07  01  --  30  42   --  42  --  --  --   -------  -  -
0018   00080  --  --  31  0F  66  31   --  --  --  --  --   -------  -  -
0019   00040  --  --  --  36  15  64   64  --  --  01  --   0------  -  -
0020   00050  --  --  --  78  42  EA   --  --  EA  --  --   -------  -  -
0021   00060  --  --  --  40  --  52   52  --  --  --  --   ---1--   -  -
0022   00070  --  06  01  --  64  DC   --  DC  --  --  --   -------  -  -
0023   00080  --  --  65  0F  EA  65   --  --  --  --  --   -------  -  -
0024   00040  --  --  --  36  52  C3   C3  --  --  01  --   0------  -  -
0025   00050  --  --  --  78  DC  33   --  --  33  --  --   -------  -  -
0026   00060  --  --  --  40  --  89   89  --  --  --  --   ---1--   -  -
0027   00070  --  05  01  --  C3  78   --  78  --  --  --   -------  -  -
0028   00080  --  --  C2  0F  33  C2   --  --  --  --  --   -------  -  -
0029   00040  --  --  --  36  89  19   19  --  --  01  --   0------  -  -
0030   00050  --  --  --  78  78  E6   --  --  E6  --  --   -------  -  -
0031   00060  --  --  --  40  --  35   35  --  --  --  --   ---1--   -  -
0032   00070  --  04  01  --  19  CA   --  CA  --  --  --   -------  -  -
0033   00080  --  --  18  0F  E6  18   --  --  --  --  --   -------  -  -
0034   00040  --  --  --  36  35  6E   6E  --  --  01  --   0------  -  -
0035   00050  --  --  --  78  CA  DF   --  --  DF  --  --   -------  -  -
0036   00060  --  --  --  40  --  FE   FE  --  --  --  --   ---1--   -  -
0037   00070  --  03  01  --  6E  4E   --  4E  --  --  --   -------  -  -
0038   00080  --  --  6F  0F  DF  6F   --  --  --  --  --   -------  -  -
0039   00040  --  --  --  36  FE  90   90  --  --  01  --   0------  -  -
0040   00050  --  --  --  78  4E  5E   --  --  5E  --  --   -------  -  -
0041   00060  --  --  --  40  --  0C   0C  --  --  --  --   ---1--   -  -
0042   00070  --  02  01  --  90  C2   --  C2  --  --  --   -------  -  -
0043   00080  --  --  91  0F  5E  91   --  --  --  --  --   -------  -  -
0044   00040  --  --  --  36  0C  72   72  --  --  01  --   0------  -  -
0045   00050  --  --  --  78  C2  7F   --  --  7F  --  --   -------  -  -
0046   00060  --  --  --  40  --  BF   BF  --  --  --  --   ---1--   -  -
0047   00070  --  01  01  --  72  B3   --  B3  --  --  --   -------  -  -
0048   00080  --  --  73  0F  7F  73   --  --  --  --  --   -------  -  -
0049   00040  --  --  --  36  BF  84   84  --  --  01  --   0------  -  -
0050   00050  --  --  --  78  B3  2D   --  --  2D  --  --   -------  -  -
0051   00060  --  --  --  40  --  38   38  --  --  --  --   ---1--   -  -
0052   00070  --  00  01  --  84  91   --  91  --  --  --   -------  -  -
0053   00080  --  --  85  0F  2D  85   --  --  --  --  --   -------  -  -
0054   00040  --  --  --  36  38  CD   CD  --  --  01  --   0------  -  -
0055   00050  --  --  --  78  91  D4   --  --  D4  --  --   -------  -  -
0056   00060  --  --  --  40  --  C2   C2  --  --  --  --   ---0--   -  -
0057   00070  --  1F  01  --  CD  DA   --  DA  --  --  --   -------  -  -
0058   00080  --  --  CC  0F  D4  CC   --  --  --  --  --   -------  -  -
0059   00090  --  0F  --  --  --  --   --  --  --  --  --   -------  -  -
0060   00100  --  0E  --  36  C2  A4   --  --  --  --  A4   -------  -  -
0061   00110  --  0D  --  78  DA  B4   --  --  --  --  B4   -------  -  -
0062   00120  --  0C  --  40  --  7D   --  --  --  --  7D   -------  -  -
0063   00130  --  --  --  --  --  6D   --  --  --  --  6D   -------  -  -
```

FIG.8 AN EXAMPLE OF TRACING RESULT

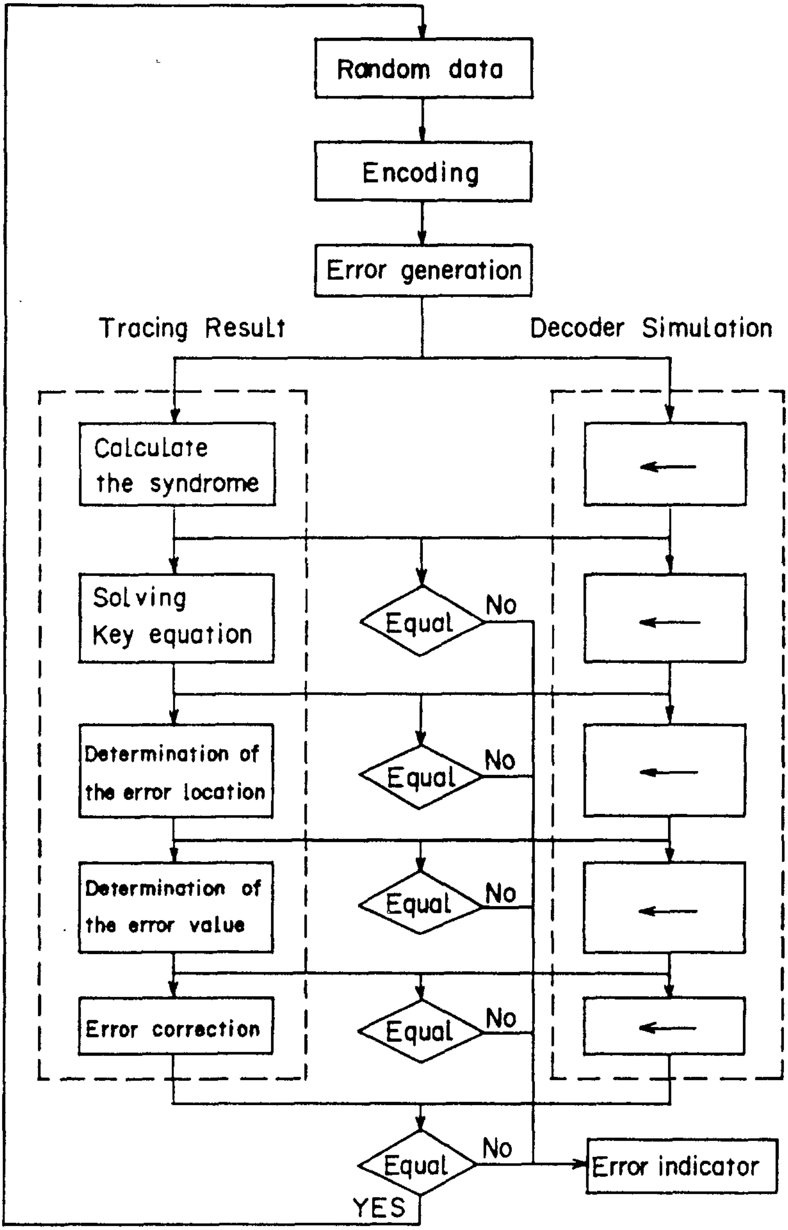


FIG.9 PROGRAM CHECK FLOW

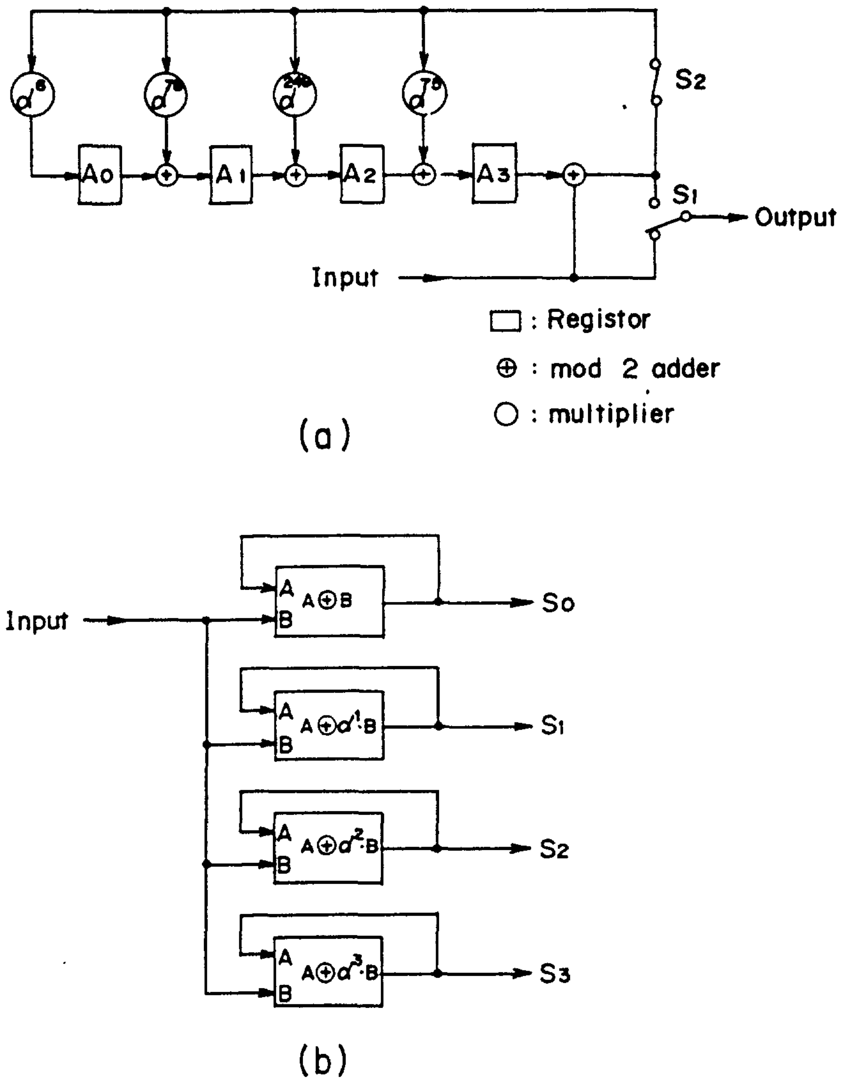
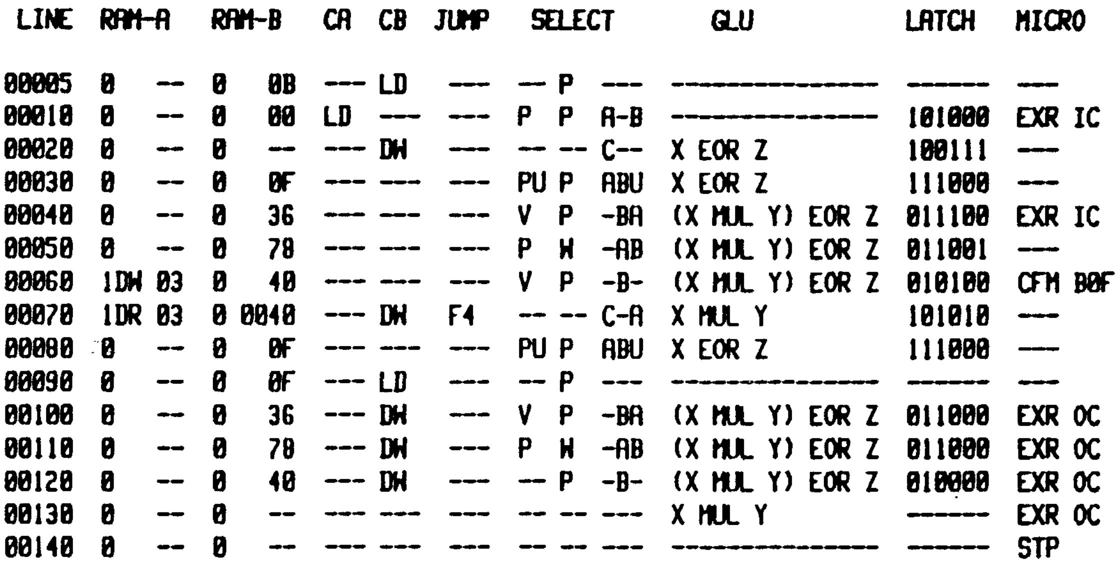


FIG. 10 (a) ENCODER
(b) SYNDROME GENERATOR
BLOCK DIAGRAM



```
 LINE  RAM-A    RAM-B    CA  CB  JUMP   SELECT       GLU             LATCH   MICRO

00005  0    --  0    0B  --- LD   ---   -- P  ---  ---------------  ------  ---
00010  0    --  0    00  LD  ---  ---   P  P  A-B  ---------------  101000  EXR IC
00020  0    --  0    --  --- DW   ---   -- -- C--  X EOR Z          100111  ---
00030  0    --  0    0F  --- ---  ---   PU P  ABU  X EOR Z          111000  ---
00040  0    --  0    36  --- ---  ---   V  P  -BA  (X MUL Y) EOR Z  011100  EXR IC
00050  0    --  0    78  --- ---  ---   P  W  -AB  (X MUL Y) EOR Z  011001  ---
00060  1DW 03   0    40  --- ---  ---   V  P  -B-  (X MUL Y) EOR Z  010100  CFM B0F
00070  1DR 03   0 0040   --- DW   F4    -- -- C-A  X MUL Y          101010  ---
00080  0    --  0    0F  --- ---  ---   PU P  ABU  X EOR Z          111000  ---
00090  0    --  0    0F  --- LD   ---   -- P  ---  ---------------  ------  ---
00100  0    --  0    36  --- DW   ---   V  P  -BA  (X MUL Y) EOR Z  011000  EXR OC
00110  0    --  0    78  --- DW   ---   P  W  -AB  (X MUL Y) EOR Z  011000  EXR OC
00120  0    --  0    40  --- DW   ---   -- P  -B-  (X MUL Y) EOR Z  010000  EXR OC
00130  0    --  0    --  --- ---  ---   -- -- ---  X MUL Y          ------  EXR OC
00140  0    --  0    --  --- ---  ---   -- -- ---  ---------------  ------  STP
```

FIG.11 ENCODER PROGRAM

```
 LINE  RAM-A   RAM-B    CA  CB  JUMP   SELECT       GLU                 LATCH   MICRO

00005  !INITIALIZE
00010  0   --  0   00   LD  ---  ---   P  P  A-B  ---------------   101000  ---
00020  0   --  0   0F   --- LD   ---   -- P  ---  X EOR Z           000111  ---
00030  0   --  1DW 01   --- ---  ---   -- W  ---  ---------------   ------  EXR IC
00040  0   --  1DW 02   --- ---  ---   -- W  ---  ---------------   ------  ALF
00050  0   --  1DR 00   --- DW   ---   -- -- CCB  ---------------   111000  ---
00055  ! CALCULATION OF SYNDROME
00060  0   --  1DR 01   --- ---  ---   -- -- -CB  (X EOR Z) MUL Y   011100  EXR IC
00070  1DW 00  0   --   --- ---  ---   V  -- -CU  (X EOR Z) MUL Y   011100  CFM B0T
00080  1DW 01  0   --   --- ---  ---   V  W  -CB  (X EOR Z) MUL Y   011001  ALF
00090  1DR 00  0 0140   --- DW   F4    V  -- CCA  (X EOR Z) MUL Y   111100  ---
00100  1DR 01  0   --   --- ---  ---   -- -- -CA  (X EOR Z) MUL Y   011010  ---
00110  0   --  1DW 00   --- ---  ---   -- W  -CU  (X EOR Z) MUL Y   011010  EXR IC
00120  0 0060  1DW 01   --- ---  UNC   V  W  -CA  (X EOR Z) MUL Y   011001  ALF
00130  0   --  1DR 00   --- DW   ---   -- -- CCB  (X EOR Z) MUL Y   111010  ---
00140  1DR 01  0   --   --- ---  ---   -- -- --A  X EOR Z           001010  ---
00150  0   --  1DW 00   --- ---  ---   -- W  --U  X EOR Z           001010  ---
00160  0   --  1DW 01   --- ---  ---   V  W  --A  X EOR Z           001010  ---
00170  0   --  1DW 02   --- ---  ---   -- W  ---  X EOR Z           000010  ---
00180  0   --  1DW 03   --- ---  ---   -- W  ---  ---------------   ------  STP
```

FIG.12 SYNDROME GENERATOR PROGRAM