THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 51, No. 2, February, 1972
Printed in U.S.A.

The Design and Embodiment of Magnetic Domain Encoders and Single-Error Correcting Decoders for Cyclic Block Codes

By S. V. AHAMED

(Manuscript received July 19, 1971)

This paper explores the possibilities of accomplishing the functional requirements of encoders and single-error correcting decoders for cyclic block codes using the inherent properties of magnetic domains. Typical designs and embodiments of such encoders and decoders are presented with field access propagations for moving the magnetic domains.

I. INTRODUCTION

The properties of magnetic domains have been studied by A. H. Bobeck,[1–3] by A. J. Perneski,[4] by U. F. Gianola,[3] and by A. A. Thiele.[5,6] The applications of such magnetic domains for storage and logic are described by A. H. Bobeck, R. F. Fischer, and A. J. Perneski,[7,8] and by P. I. Bonyhard, et al.[9] This paper proposes the possible applications of these results to the construction of encoders and single-error correcting decoders for cyclic block codes.

Single-error correcting codes were introduced by R. W. Hamming in 1950.[10] In 1960, R. C. Bose and D. K. Ray-Chaudhuri[11] formulated a class of multiple-error correcting cyclic codes. W. W. Peterson[12] has presented a variety of logic circuits for encoders and decoders. These circuits conventionally employ semiconductor logic elements. Such circuits are discussed in some detail by R. W. Lucky, J. Salz, and E. J. Weldon, Jr.,[13] and by E. R. Berlekamp.[14]

The encoders proposed in this paper are for cyclic block codes and the decoders are limited to single-error correcting decoders for such codes. In general, these codes constitute a set of BCH codes named after Bose, Chaudhuri, and A. Hocquenghem.[15] The paper is divided into four parts. Part A provides an insight into the fundamentals necessary for a qualitative understanding of magnetic domain functions. Part B deals with encoders for cyclic block codes and Part C deals with single-error correcting decoders for such codes. Part D discusses the various types of magnetic materials suitable for embodiments and their characteristics. Each of the two parts B and C is divided into two sections; section 1 introduces the fundamentals of encoding and decoding while section 2 leads into the conversion of conventional encoding and decoding to serial encoding and decoding, and describes these functions with magnetic domains. The expert in magnetic domain devices may skip Part A and the expert in coding theory may skip the first sections of Parts B and C.

(PART A)

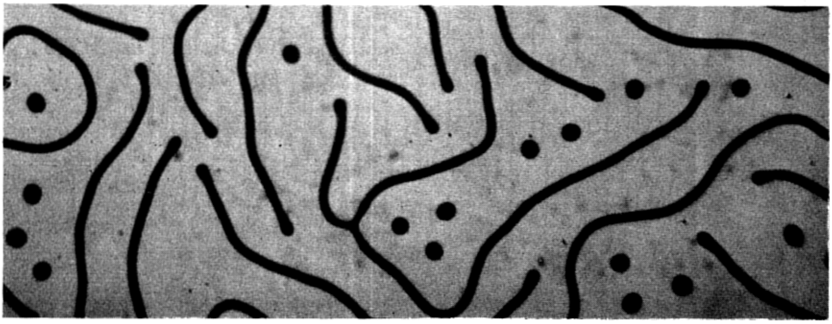
II. MAGNETIC DOMAINS AND THEIR FUNCTIONS

2.1 *Introduction to Orthoferrites and Domains*

Rare earth orthoferrites have a formula $RFeO_3$ where R is a rare earth. Very thin platelets of the orthoferrite crystals are prepared so that the appropriate crystalline axis (generally the 001 or *c* axis) is normal to the surface of the platelet. Magnetic domains with their direction or magnetization normal to the surface of such platelets may be observed by Faraday effect. Such domains may also be observed (Fig. 1a) in very thin epitaxial garnet films deposited on suitable substrates. When these domains are subjected to a bias field opposing the magnetic moment enclosed within them, they shrink (Fig. 1b) to microscopic and submicroscopic sizes and are cylindrical in shape. Such cylindrical magnetic domains, sometimes called bubbles, generally

(a)

Fig. 1a—Magnetic domains in a typical epitaxial film 5 to 8 microns deep, deposited on Gadolinium–Gallium–Garnet (GGG) substrate 20 to 40 mils thick. Magnification is 340.



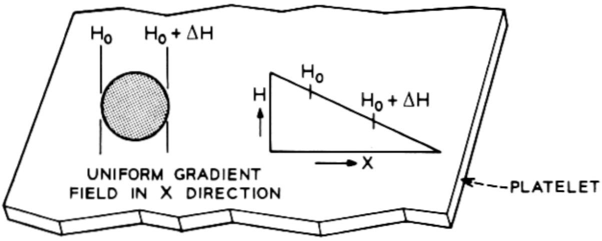
(b)

Fig. 1b—Formation of "bubbles" from magnetic domains at a bias field of 30 Oe in the same material used in Fig. 1a. Magnification is 340.

are a few microns in diameter and are stable under proper bias field conditions. Bubbles may be used to store information and to carry out certain elementary logical functions.

2.2 *Propagation of Bubbles*

Bubbles respond to bias field gradients in the plane of the platelet (or film) hosting them by moving in a direction which tends to minimize the net energy. A bubble of diameter d located in a uniform gradient field would tend towards the position of reduced bias (Fig. 2a). Bubble velocities yielding a bit rate of over two or three megacycles have been achieved in selected magnetic materials. There are two basic methods of providing such an inplane field gradient. The first method depends on a current in a conductor loop which produces a field to attract the neighboring bubble directly beneath a loop formed by a conductor (Fig. 2b). This method is called "conductor propagation" since a



(a)

Fig. 2a—A cylindrical domain located in a uniform gradient field.

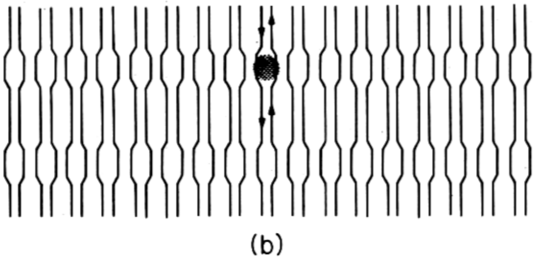


Fig. 2b—Conductor propagation of bubbles.

sequence of bubble positions may be propagated by exciting a series of conductor loops wired to carry current pulses. The second method depends on the alternating magnetization of a patterned soft magnetic overlay embedded on the surface of the platelet (Fig. 2c). The magnetization is imposed by a rotating inplane magnetic field generated by a set of two coils carrying an alternating current and surrounding the platelet with their axes in its plane. This method is called the "field access propagation" and each of the bubbles is propagated to the next pattern in the overlay during one cycle of the exciting current in the surrounding coils.

Field access propagation is more suitable for constructing magnetic domain encoders and decoders, even though it is possible to construct these devices with conductor propagation. Storage, propagation, and the synchronization of incoming data with the outgoing data may all be accomplished by one clock driven at one frequency which is a multiple of the transmission rate. For this reason, only the embodiments of encoders and decoders with field access propagation will be discussed in this paper.

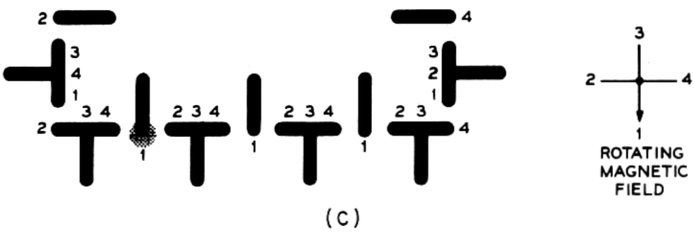


Fig. 2c—Field access propagation of bubbles.

2.3 *Bubble Functions*

2.3.1 *Generation of Bubbles*

Bubbles are generated from an original source bubble. The source bubble rotates around the periphery of a disk of soft magnetic material and when subjected to the localized field of a properly placed current loop it is duplicated (Fig. 3). One section is led away from the generator

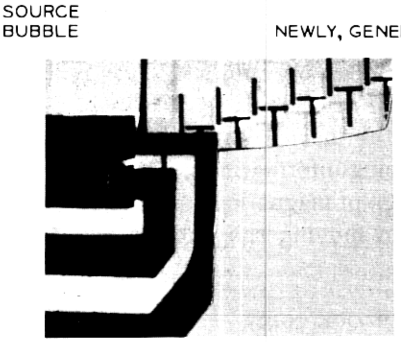


Fig. 3—Generation of bubbles in field access propagation.

and the other section keeps rotating. In most cases the bubbles are generated only when subjected to the field of a current, which is generally controlled by the information bits, or by readings of sensors in a circuit.

2.3.2 *The exclusive-or operation*

This function is accomplished by the mutual repulsion of two bubbles when they are brought in close proximity. In Fig. 4 any one of the two input bubbles A or B finds its way to the output in the absence of a repulsive force due to the other input bubble. Two input bubbles mutually repel themselves into two annihilators. Such an annihilator operates by merging the incoming bubble with a bubble of its own and the diameter of the bubble in the annihilator remains as it was before the merging of the incoming bubbles.

(PART B)

Overview

The basic vehicle chosen for introducing the principles of encoding is the single-error correcting (7,4)* cyclic block Hamming code. The

* The notation is explained in Section 3.1.

principles are then extended to a (31,21) cyclic block code. The code chosen to demonstrate the feasibility of the designs and the embodiments of magnetic domain encoders is (30,20) shortened block code. It is derived from the original (31,21) code. This choice, even though it is inherently a double-error correcting code, facilitates the presentation of serial encoding with field access propagation. The generality of the embodiment for another code is also presented.

In the design of encoders and decoders, time plays a critical part and it becomes necessary to choose a unit of time for any given code. In a (n,k) code, if the incoming information is received at the rate of k bits every P seconds, then the outgoing information is relayed at the rate of n bits every P seconds. If t is defined as $P/(n \times k)$ seconds, then the average interval between the incoming information is $(n \times t)$ and the interval between the outgoing information is $(k \times t)$ seconds. As it will become evident in the design of magnetic domain encoders and decoders, t plays a dominant role in moving the bubbles from one location to the next.

III. ENCODERS FOR CYCLIC BLOCK CODES

(PART B, SECTION 1)

3.1 *Cyclic Block Codes and their Construction**

Block codes constitute a set of codes in which a binary block of k

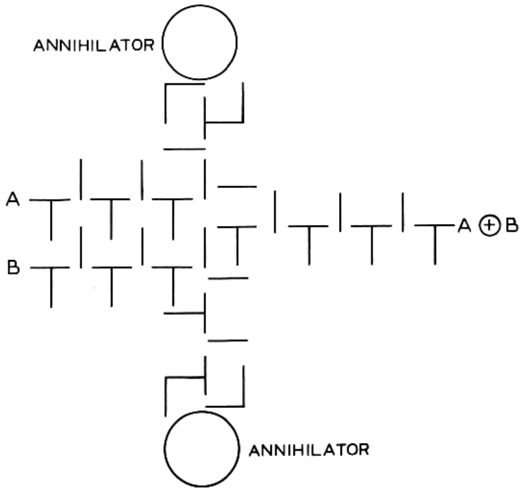


Fig. 4—Exclusive-or operation in field access propagation.

* This topic is discussed in Refs. 12, 13, and 14.

information bits has a binary block of $(n - k)$ parity bits appended to it, thus constituting a (n,k) block code. The n-bit binary cyclic block is represented as a polynomial $c(X)$ as follows.

Let the n-bit binary string be 1010001. The polynomial representation would be

$$c(X) = X^6 + X^4 + X^0 = X^6 + X^4 + 1 \tag{1}$$

corresponding to unity in the first, third, and seventh binary positions. Cyclic block codes have the attractive property that if coefficients of $c(X)$ are cyclically shifted, the new polynomial also represents a code word. For instance, cyclically shifting the coefficients of $c(X)$ once, yields $(X^5 + X + 1)$ which represents another code word.

Consider a new polynomial,

$$g(X) = X^3 + X^2 + 1, \tag{2}$$

which is four binary bits long. If $c(X)$ is divided by $g(X)$ as

$$
\begin{array}{r l}
 & \overline{X^3 + X^2 + 0 + 1} \\
X^3 + X^2 + 0 + 1 \,| & X^6 + 0 + X^4 + 0 + 0 + 1 \\
 & \underline{X^6 + X^5 + 0 + X^3} \\
 & \quad X^5 + X^4 + X^3 + 0 \\
 & \quad \underline{X^5 + X^4 + 0 + X^2} \\
 & \qquad 0 + X^3 + X^2 + 0 \\
 & \qquad \underline{0 + 0 \;\; + 0 \;\; + 0} \\
 & \qquad\quad X^3 + X^2 + 0 + 1 \\
 & \qquad\quad \underline{X^3 + X^2 + 0 + 1} \\
 & \qquad\qquad\quad 0 \; + 0 + 0,
\end{array}
$$

the remainder is three binary zeros. When polynomials obtained by cyclically shifting the coefficients of $c(X)$ once, twice, etc., are divided by $g(X)$, the three-bit remainders obtained are always zero. For each cyclic code there exists such a polynomial $g(X)$ which divides every codeword. This polynomial is called the generator of the code.

Now consider a new polynomial $d(X)$ which corresponds to the first four bit positions of $c(X)$ yielding

$$d(X) = X^6 + X^4. \tag{3}$$

If $d(X)$ is divided by $g(X)$ the remainder corresponds to the polynomial

$$r(X) = 0 \cdot X^2 + 0 \cdot X + 1 = 1, \tag{4}$$

corresponding to the last three bits of the polynomial $c(X)$, since $g(X)$ divides $c(X)$ completely. If the first four bits of $c(X)$ were to denote information bits of a code, then, the last three bits may be thought of as the parity bits, and are in general obtained by dividing a data polynomial $d(X)$ by the generator $g(X)$ and calculating the remainder.

The paper by Bose and Chaudhuri[11] has proved that a large number of codes may be generated by various choices of n and k, provided a generator polynomial $g(X)$ exists for the particular combination of n and k. The value of n is initially limited to $(2^m - 1)$ where m is an integer number. The series of polynomials $g(X)$ for each value of n are readily available in any standard textbook in coding theory (see Ref. 12 or 13). One such value of n is 31 (i.e., $2^5 - 1$) and one of the polynomials for $g(X)$ is

$$g(X) = X^{10} + X^9 + X^8 + X^6 + X^5 + X^3 + 1. \tag{5}$$

The highest degree of the remainder $r(X)$ in the division of a polynomial $d(X)$ by $g(X)$ is always one less than the degree of divisor $g(X)$. In this case, the highest degree of $r(X)$ is 9 and is 10 bits long. Hence the cyclic block code constructed with $n = 31$ and the prechosen value of $g(X)$ has 10 parity bits leading to a (31,21) code. It is however possible to reduce both n and k by a selected number and obtain shortened block codes. For example, if the first bit of an original (31,21) code is eliminated by considering it as being always zero, then a (30,20) code is obtained yielding 10 parity bits for every 20 information bits and the rate corresponding to the ratio of k to n is 2/3.

3.2 *The Function of Encoding for Cyclic Block Codes*

The encoder receives information $d(X)$ in blocks from the data source and yields the code word $c(X)$ in blocks. The two subfunctions are

(*i*) Divide the incoming data string $d(X)$ by the generator function $g(X)$, and
(*ii*) Append the remainder after the division to the incoming data string.

These subfunctions are commonly accomplished by semiconductor electronic circuitry in conventional encoders. The division in Section 3.1 has four steps. During the first step of the division cycle the nonzero terms of $g(X)$ are added (by an exclusive-or operation) to the appropriate terms of the data polynomial $d(X)$. At the successive steps of the division cycle, the partial remainder from the earlier step is treated the same way, and the nonzero terms of $g(X)$ are added (by the exclusive-or function) to the appropriate terms, provided the highest order of the partial remainder is a nonzero quantity. When a zero quotient is encountered in the highest order (such as the third step of the division), then a set of zeros is added (by an exclusive-or function) to the partial remainder.

In the electronic circuitry these functions may be explicitly accomplished. The steps in the well-known (but not frequently used) encoder,* shown in Fig. 5a, are as follows.

Step 1:	Switch	S_1	S_2	S_3
	Position	1	1	1

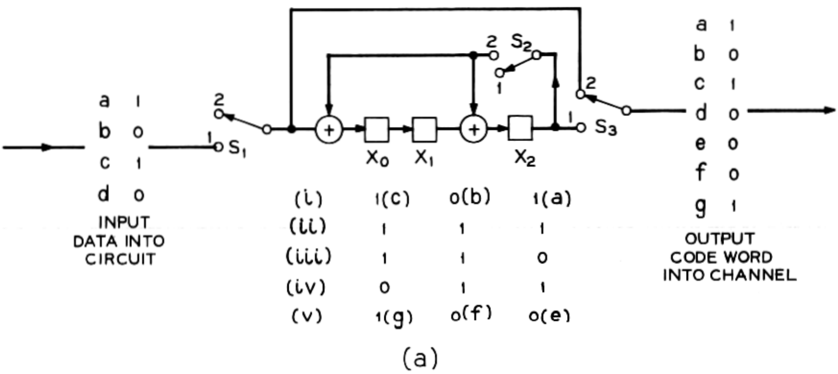


Fig. 5a—Encoding for a (7,4) block cyclic code with generator function $g(X) = 1 + X^2 + X^3$.

The first three data positions 101 of $d(X)$ in (3) are shifted into the encoder and transmitted into the channel [line (i) in Fig. 5a].

Step 2:	Switch	S_1	S_2	S_3
	Position	1	2	2

The last data bit, i.e., 0, is shifted into the register and transmitted into the line. The contents of the shift register are shown in line (ii). (Also see the partial remainder after the first step of the division cycle in Section 3.1.)

Step 3:	Switch	S_1	S_2	S_3
	Position	2	2	2

* It will be seen that this type of encoder will present certain operating advantages with magnetic domain configurations in which storage is quite inexpensive as compared to semiconductor configurations.

The contents of the shift register are shifted three times (corresponding to the three remaining steps of the four-step division cycle). The contents of the shift register are shown by (iii), (iv), and (v).

Step 4:	Switch	S_1	S_2	S_3
	Position	2	1	1

The contents of the shift register are emptied into the channel and these correspond to the parity bits $r(X)$ in (4).

A more commonly used configuration of the encoder arrangement is shown in Fig. 5b. Four data bits are shifted with switches S_1, S_2, and S_3 in position 2. The switches are moved down to position 1 and the contents of the shift register are emptied into the channel. The lines a, b, c, and d in Fig. 5b indicate the contents of the register as the data bits corresponding to $d(X)$ in (3) are received.

It is to be observed here that the arrangement in Fig. 5b necessitates that the two exclusive-or functions be done serially between the arrival of data bits, whereas the arrangement in Fig. 5a requires that the two exclusive-or functions be accomplished simultaneously. In the magnetic domain technology this consideration makes the configuration of Fig. 5a more favorable for the implementation.

Encoders for various codes are similarly constructed. The location of the exclusive-or gates is determined by the nonzero terms in the generator polynomial $g(X)$ exclusive of the highest-degree term. Figures 5c and 5d indicate the conventional encoder arrangements for the (30,20) shortened block code discussed earlier with $g(X)$ in (5). The complete encoder also adjusts for the difference of rate between the

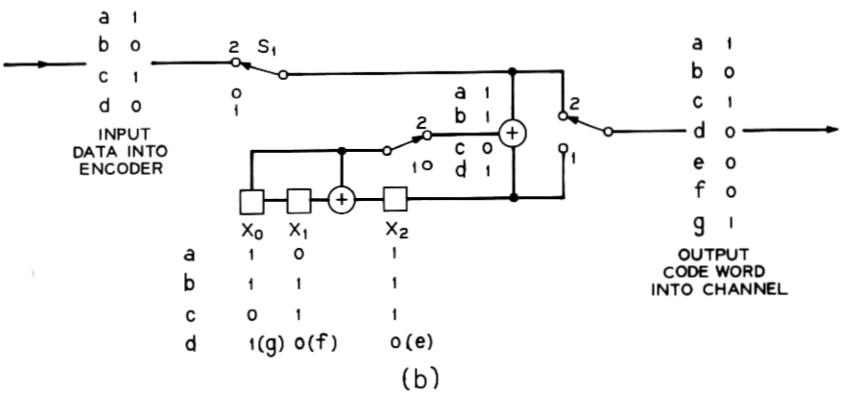


Fig. 5b—Conventional encoder for the (7,4) block cyclic code with the same generator used in Fig. 5a.

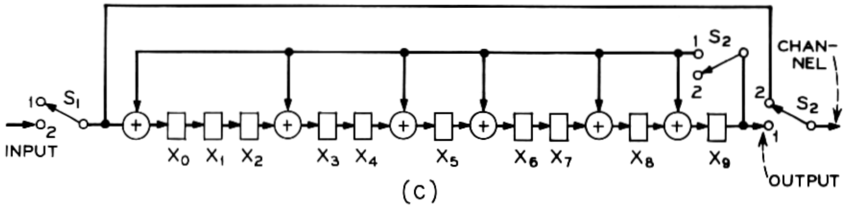


Fig. 5c—An encoding arrangement for (30,20) cyclic block code with generator function

$$g(X) = 1 + X^3 + X^5 + X^6 + X^8 + X^9 + X^{10}.$$

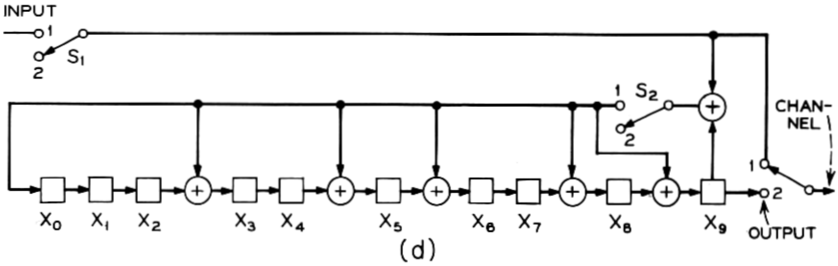


Fig. 5d—Conventional encoder configuration for the (30,20) code with same generator used in Fig. 5c.

arrival rate at the input of the encoder and its output into the channel. In this case the input rate in the encoder is two-thirds the output rate.

The generator $g(X)$ in (5) has seven terms. During each step of the division cycle the highest order term is eliminated from the partial remainder in the shift register. This leads to an unconditional zero coefficient for the term to which X^{10} is added. This fact may be used to limit the number of terms in $g(X)$ to six terms (excluding the highest-order term), provided the highest-order term is eliminated from the partial remainder after sensing its value.* Under such conditions the six remaining terms may be written as

$$g'(X) = 1 + X^3 + X^5 + X^6 + X^8 + X^9. \tag{6}$$

(PART B, SECTION 2)

3.2.1 *Serial Arrangement of Encoders for Cyclic Block Codes*

In the conventional configurations (Figs. 5c and 5d), the output of the rightmost stage feeds back into six different exclusive-or gates

* The presence of one in the highest-order position requires that the other six terms be added (by exclusive-or function) to the corresponding terms in the partial remainder.

corresponding to the nonzero terms of $g(X)$. Alternatively, the information may be fed back at one location with one exclusive-or gate but at six different instants of time. Each step of the 20*-step division cycle is effectively performed by circulating the partial remainder through this gate. The input to the gate is dictated by the nonzero terms of $g'(X)$. A configuration incorporating such a serial feedback arrangement is shown in Fig. 6a. The switch S_a is designed to respond to the contents of x_{10}, closing only if the content is one. The contents of the shift register $g'(X)$ are initialized to 1101101001 corresponding to the X^9, X^8, X^6, X^5, X^3, 1 terms of the function $g'(X)$ in (6). The contents of the shift register $g'(X)$ are circulated in synchronism with the main shift register SR. The circulation time of both registers is the time between the arrival of bits in the incoming data stream.

The operation of this type of encoder after emptying its contents is as follows:

Step 1:	Switch	S_1	S_2	S_2'	S_3
	Position	1	2	2	1

The first 10 bits of a data block are shifted into the main shift register SR.

Step 2:	Switch	S_1	S_2	S_2'	S_3
	Position	1	1	2	1

The shift register is shifted once more so that the highest-order bit is in x_{10} and the 11th bit of the data block enters position x_0 simultaneously.

Step 3:	Switch	S_1	S_2	S_2'	S_3
	Position	1	2	1	1

The shift register is completely circulated once.

Step 4:	Switch	S_1	S_2	S_2'	S_3
	Position	1	1	2	1

The highest-order bit is entered in x_{10} as in step 2 and the 12th bit of data enters position x_0. The process in steps 3 and 4 is repeated 10 times (i.e., 8 more times). After the 20th data bit enters the shift register, the switch S_1 is moved to position 2 and the shift register is circulated 10 more times as in steps 2 and 3. The division is now complete.

Step 5:	Switch	S_1	S_2	S_2'	S_3
	Position	2	2	2	2

* i.e., 30–10 or k, the number of information bits in the block.

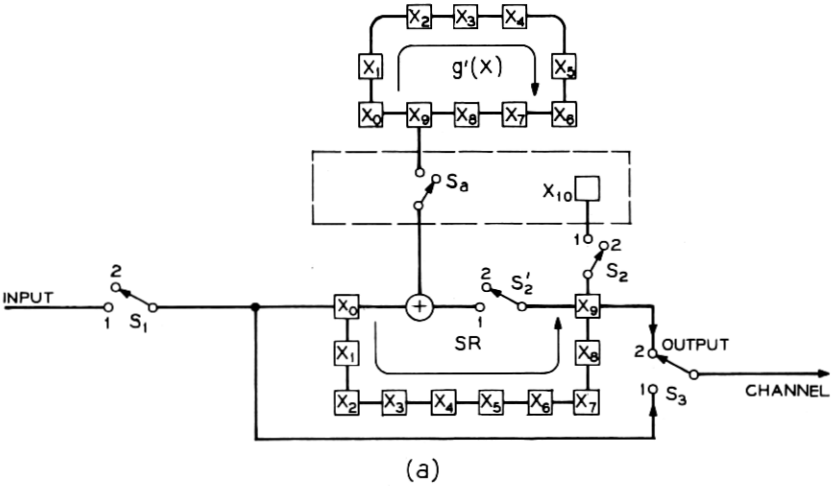


Fig. 6a—Serial encoding for the (30,20) code.

The parity bits are emptied into the channel. The process is repeated for the next data block by returning to step 1.

3.2.2 *Complete Encoder with Serial Arrangement for Cyclic Block Code*

Figure 6b shows a complete encoder. The incoming data ($k = 20$ information bits) arrives uniformly at the encoder and coded information ($n = 30$ bits consisting of 20 information and 10 parity bits) is recovered uniformly. The operation of the switch S_a is explained earlier. Only one of three poles of switch S_b is closed at any given instant of time. Coded information is emptied out of d′, d″, or d_p one bit every $20t$ seconds. The data-stores d′ and d″ store the first and second 10 bits, and d_p stores the parity bit. The data-store d_i holds the first 10 bits of any block on an interim basis while the main shift register SR is calculating the parity bits of the previous data block.

When the register is full, the contents of d_i are moved into both the main shift register SR and d′ within the $30t$ seconds preceding the arrival of the next data bit. The shift is synchronized with moving the parity bits from SR to d_p with S_5 in position 2. The arrival of the 11th bit is synchronized with the movement of the first bit into x_{10}, thus emptying the location x_0 in SR for this 11th bit. The second 10 bits arrive at location x_0 of SR via switch S_1 and are also entered in d″. The circulation SR and division with $g(X)$ continues 20 times. The data-store d_i would then have emptied the first 10 bits of the next data

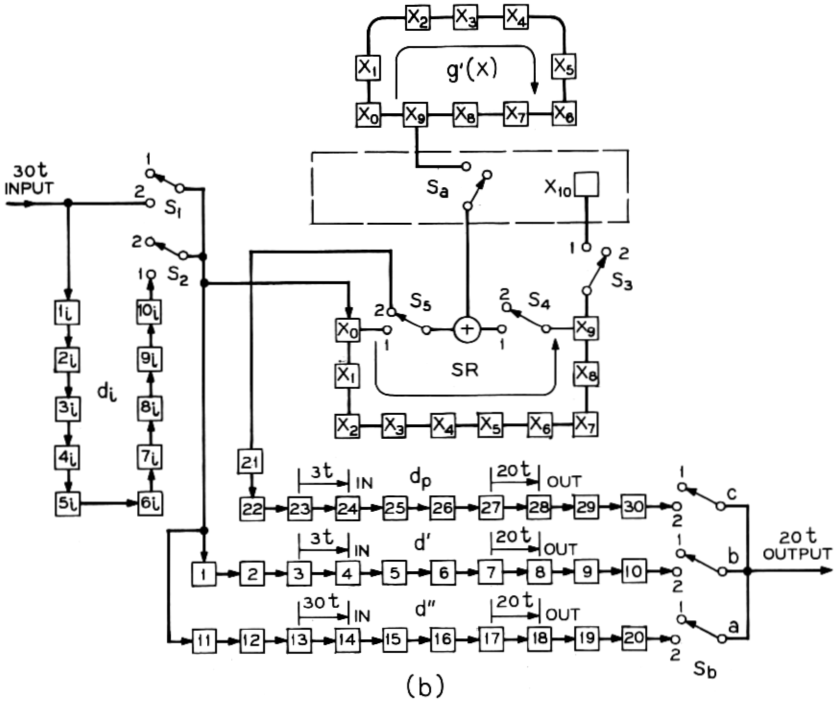


Fig. 6b—A complete serial encoder for the (30,20) code.

block into SR and d_p would have the parity bits for the data just processed. The cycle can be repeated indefinitely provided the stored d′, d″, and d_p have been emptied into the transmission channel at appropriate times. The operation of switches a, b, and c meet this requirement. The timing diagram of the encoder is shown in Fig. 6c. The incoming data is shown in line (i) and the outgoing information is shown on line (iii). The data-stores d′ and d_p shift in during a $30t$-second interval and shift out into the channel one bit every $20t$ seconds. The data-store d″ shifts in one bit every $30t$ seconds and uniformly shifts out one bit every $20t$ seconds.

3.3 *Magnetic Domain Encoders for Cyclic Block Codes with Field Access Propagation*

In field access propagation all the bubbles in the region are propagated by one pitch (or period) during one cycle time of the rotating magnetic field. It is advantageous to equate the cycle time of this magnetic field with t defined earlier.

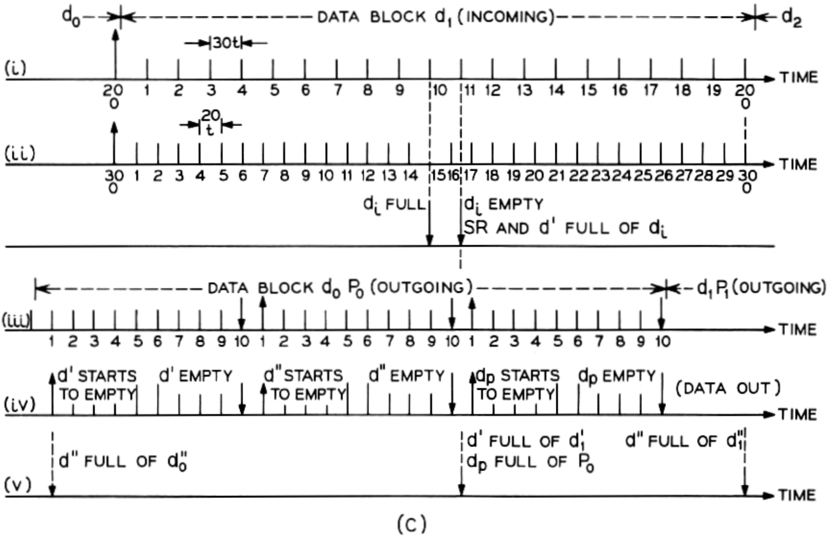


Fig. 6c—Timing diagram for encoder shown in Fig. 6b.

The encoder arrangement is shown in Fig. 7. The incoming data pulses generate bubbles at the information bubble generator. These are accumulated in loop 1 at consecutive periods since there are 29 periods and the incoming data arrives every $30t$ seconds. The channeling gate g_1 opens every 20 circulations* to permit a sequence of 20 bubble positions to enter the duplicator D. The data is circulated in loop 3. Loop 2 performs one step of the 20-step division cycle every circulation. The sensor S_g reads the leading bubble position every $30t$ seconds and controls the generator G_g to inject a series of bubbles corresponding to the nonzero terms of the generator $g(X)$ in the exclusive-or gate. A string of bubbles 11101101001, corresponding to X^{10}, X^9, X^8, X^6, X^5, X^3, 1 terms in $g(X)$, is generated if S_g has sensed a bubble. The distances between the sensor S_g, the exclusive-or gate, and generator G_g are adjusted so that the bubble corresponding to the X^{10} position of $g(X)$ arrives into the exclusive-or gate in synchronism with the leading bubble position that S_g sensed. After 20 circulations, the 10 parity bits are left in loop 2. The parity and data bits are channeled into loops 4 and 5 by the action of the channeling gates (Ref. 9) g_2 and g_3 respectively.

The code word (data and parity) is retrieved and transmitted in two sections. The data is read by the sensor S_d in loop 5 every $20t$ seconds. The parity is read by the sensor S_p in loop 4 every $20t$ seconds. The

* A circulation corresponds to the contents of the loop going around once.

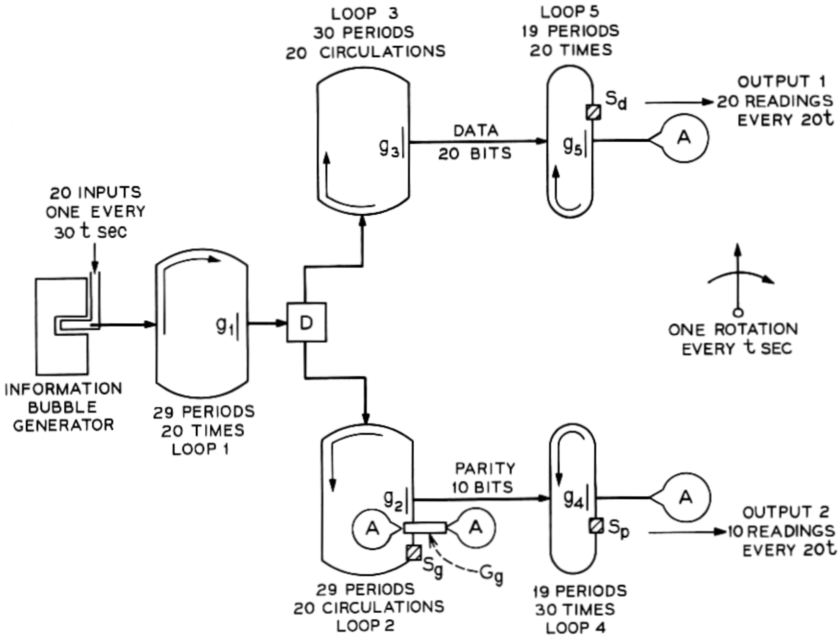


Fig. 7—Serial encoder with domains under field access propagation for the (30,20) code with $t = P/30 \times 20$.

diverting gates g_4 and g_5 function identically. Every time the sensor S_d or S_g is read, the diverting gate g_5 or g_4 diverts the bubble position read into annihilator A.

The generality of this embodiment is exemplified by another serial encoder shown in Fig. 8 for (31,26) cyclic block code. The generator function for this code is

$$g(X) = 1 + X^2 + X^5. \tag{7}$$

This encoder operates along the same principles described earlier. Such encoders cannot be constructed when the loops 1 through 5 become extremely small, and thus codes with very short block lengths cannot be easily implemented. Generally block codes with block lengths of 30 or over are well suited for such embodiment.

(PART C)

Overview

The basic vehicle chosen for introducing the principles of decoding is the single-error correcting (7,4) cyclic block Hamming code discussed earlier. The general concepts of decoding and single-error correcting

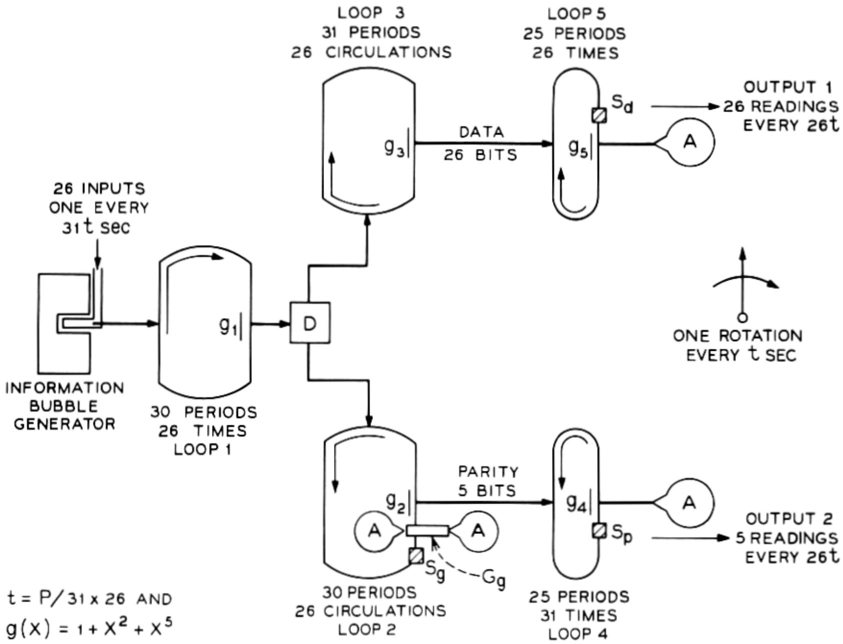


Fig. 8—Serial encoder for the (31,26) code with domains under field access propagation with $t = P/31 \times 26$.

are then extended to cyclic block codes. The particular code chosen to demonstrate the embodiment of single-error correcting decoders for cyclic block codes is the (30,20) shortened cyclic code.

For a (n,k) code, the decoders receive information from the channel at an interval of one bit every $(k \times t)$ seconds and recover the original information at an interval of one bit every $(n \times t)$ seconds. Further, the decoders discussed here detect and correct single errors in the received information. Multiple-error correcting decoders are not discussed in this paper.

IV. SINGLE-ERROR CORRECTING DECODERS FOR CYCLIC BLOCK CODES

(PART C, SECTION 1)

4.1 *Decoding of Cyclic Block Codes**

Coded information in the form of $c(X)$ in (1) is received from the channel into the decoder. The decoder recovers the original information

* This topic is discussed in Refs. 16 and 17.

bits [polynomial $d(X)$] from $c(X)$ even if one of the bits of $c(X)$ was received in error at the decoder. The function of detecting and correcting single errors for the (7,4) Hamming code is explained as follows. Let $S_1(X)$, $S_2(X)$, $\cdots$, $S_7(X)$ be the remainders obtained by dividing $X, X^2, \cdots X^7$, by $g(X)$ in (2). These polynomials may be calculated as:

$$s_1(X) = X; \qquad s_2(X) = X^2 \tag{8a; b}$$

$$s_3(X) = X^2 + 1; \qquad s_4(X) = X^2 + X + 1 \tag{8c; d}$$

$$s_5(X) = X + 1; \qquad s_6(X) = X^2 + X, \tag{8e; f}$$

and finally

$$s_7(X) = \Gamma(X) = 0 \cdot X^2 + 0 \cdot X + 1 = 1. \tag{8g}$$

Now if the received word has a single error in the second location,* then the received word $R(X)$ will differ from $c(X)$ as follows:

$$R(X) = c(X) + X^5 = X^6 + X^5 + X^4 + 1 \tag{9}$$

and the remainder [also known as the syndrome $s(X)$] obtained by dividing $R(X)$ by $g(X)$ is

$$s(X) = X + 1. \tag{10}$$

This polynomial is seen to be $s_5(X)$ from (8e) indicating that a single error in the ith location yields a syndrome corresponding to $s_{7-i}(X)$. Next consider the polynomial obtained by shifting $s(X)$ two (i.e., $7 - 5$) times,

$$X^2 \cdot s(X) = X^3 + X^2; \tag{11}$$

and the remainder, denoted by $\rho(X)$, obtained by dividing the shifted polynomial by $g(X)$ is

$$\rho(X) = 0 \cdot X^2 + 0 \cdot X + 1 = 1. \tag{12}$$

This value of $\rho(X)$ corresponds to $s_7(X)$ or $\Gamma(X)$ in (8g), since

$$X^i \cdot s(X) = X^i \cdot s_{7-i}(X), \tag{13}$$

and the remainder obtained by dividing the right side of (13) by $g(X)$ does in fact represent the remainder obtained by dividing $(X^i \cdot X^{7-i})$ or X^7 by $g(X)$ and is indeed $\Gamma(X)$. This leads to the conclusion that if the remainder obtained by dividing $s(X)$ shifted i times by $g(X)$ corresponds to $\Gamma(X)$ then the ith bit is in error. Correction is accomplished

* It should be noted that the error in the ith bit corresponds to adding the X^{7-i} term to $c(X)$.

by complementing this bit. In this case, the corrected data corresponding to the first four bits of $R(X)$ is 1010 originally represented as $d(X)$ in (3).

This reasoning may be extended to general cyclic codes and in particular to the (30,20) code. For this code, $\Gamma(X)$ is the remainder obtained by dividing X^{30} by $g(X)$ in (5) and it can be calculated as

$$\Gamma(X) = X^4 + X^5 + X^6 + X^7. \tag{14}$$

In semiconductor circuitry the division by $g(X)$ is accomplished by the top section of the decoder shown in Fig. 9 and the comparison of the contents of the register with $\Gamma(X)$ is accomplished by the AND gate. In the complete shift register two shift registers are used with one performing the comparison while the other is calculating the syndrome of the next data block.

(PART C, SECTION 2)

4.2 *Serial Decoding of Block Cyclic Codes*

In serial decoding the division is carried out by one exclusive-or gate as in serial encoding discussed in Section 3.2.1. Further, the comparison of the content of the shift register is also done serially bit by bit in contrast to the simultaneous evaluation and comparison of all the bits by the AND gate used in conventional decoders (Fig. 9).

An exclusive-or gate is used for serial comparison instead of the AND gate. In the (30,20) code the comparison cycle lasts for 10 bits

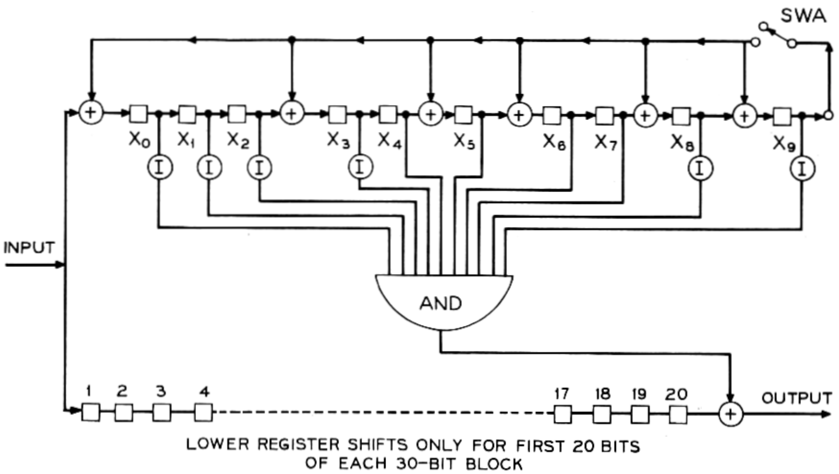


Fig. 9—Conventional single-error correcting decoder with (30,20) code.

(i.e., the number of bits in the remainder). Bits duplicated from the shift register are serially fed into an exclusive-or gate together with bits corresponding to $\Gamma(X)$. A perfect match between the two inputs yields a zero output from the exclusive-or gate for the entire interval of comparison. One or more outputs from the exclusive-or gate during the interval indicates a mismatch. This principle is used in the magnetic domain decoders discussed next.

4.3 *Single-Error Correcting Magnetic Domain Decoder for Cyclic Block Codes with Field Access Propagation*

Figure 10 shows a decoder for the (30,20) code. The operation of the decoder closely resembles the operation of the encoder shown in Fig. 7. The incoming data generates a series of bubbles at G_i. This data is led

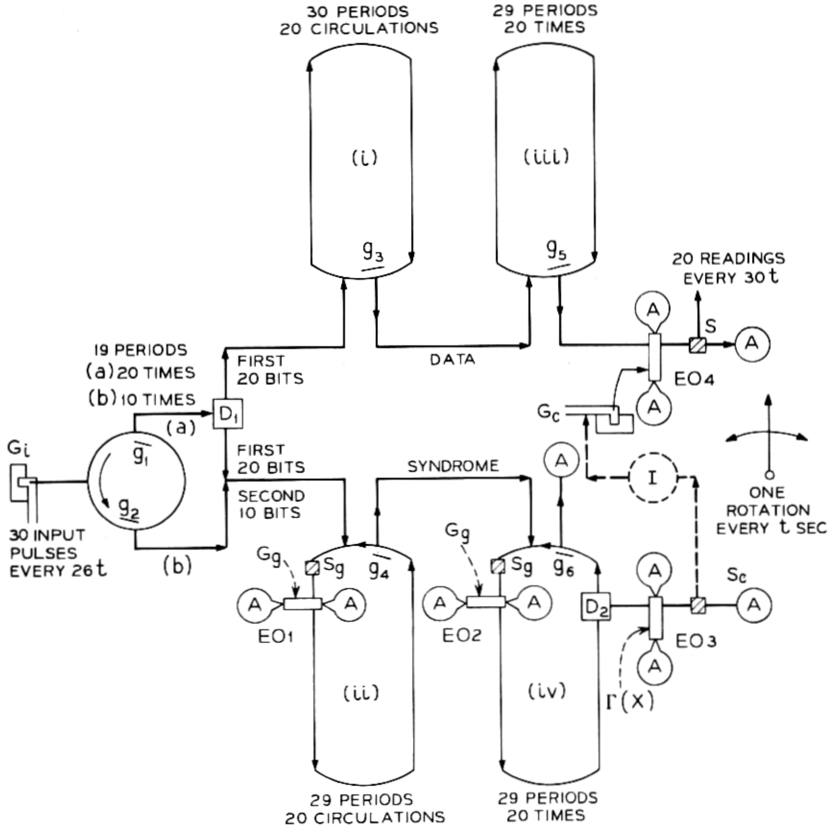


Fig. 10—Serial arrangement of single-error correcting decoder with magnetic domains under field access propagation for the (30,20) code with $t = P/30 \times 20$.

into a loop with 19 periods and two exit gates g_1 and g_2. A string of bubbles are formed in adjoining periods in the loop since the main magnetic field rotates every t seconds and the bubbles arrive every $20t$ seconds. When the first 19 bubble positions have arrived in the loop, the gate g_1 empties the bubble stream into channel (a). This stream is duplicated at D_1 and it enters loops (i) and (ii). This data is allowed to circulate in (i) while the division in (ii) progresses. The sensor S_g in (ii) senses a bubble every $30t$ seconds and directs G_g to generate a stream of bubbles into the exclusive-or gate E01 only if a bubble is sensed at the leading end of the data stream. The generator bubble stream is 11101101001 and is consistent with the nonzero coefficients of $g(X)$. A leading bubble corresponding to the 10th power of X is necessary since there is no special arrangement to eliminate the X^{10} bubble as in the conductor pattern propagation (see Fig. 6b). After the parity bits are accumulated in the first loop, they are channeled into (ii) via gate g_2. To ascertain that the last 10 bits arrive at the correct location in (ii), it is necessary to adjust the number of periods between the exit points g_1 and g_2 in the first loop.

When the division is complete [i.e., 20 circulations of (ii), each circulation accomplishing one step of the 20-step division cycle] the data and syndrome may be transferred out of (i) and (ii) by gates g_3 and g_4 respectively. Such gates have been designed and implemented for other applications (Ref. 9 and Ref. 18). In loop (iv) the syndrome is again divided by the generator function $g(X)$. The exclusive-or gate E02 performs this function. The remainder after this division is duplicated at D_2. One section circulates in (iv) and the other is compared with the remainder function $\Gamma(X)$ by the exclusive-or gate E03. A perfect match results in a zero reading of sensor S_c during the entire comparison time which lasts for $10t$ seconds. Meanwhile, the data in (iii) is also being circulated. The gate g_5 permits one leading bubble to be channelized out of the loop once every circulation into the exclusive-or gate E04. This gate receives a complementing bubble only if S_c has not sensed a bubble after comparing the contents of (iv) with the remainder function $\Gamma(X)$.

The generality of the embodiment is exemplified by another serial decoder shown in Fig. 11 for (31,26) cyclic block codes. The encoder for this code is shown in Fig. 8 and the generator function is given by equation (7). This decoder operates on a principle of serial decoding discussed earlier and the value of $\Gamma(X)$, the remainder obtained by dividing X^{31} by $g(X)$ in (7), is

$$\Gamma(X) = 0\cdot X^5 + 0\cdot X^4 + 0\cdot X^3 + 0\cdot X^2 + 0\cdot X + 1 = 1. \quad (15)$$

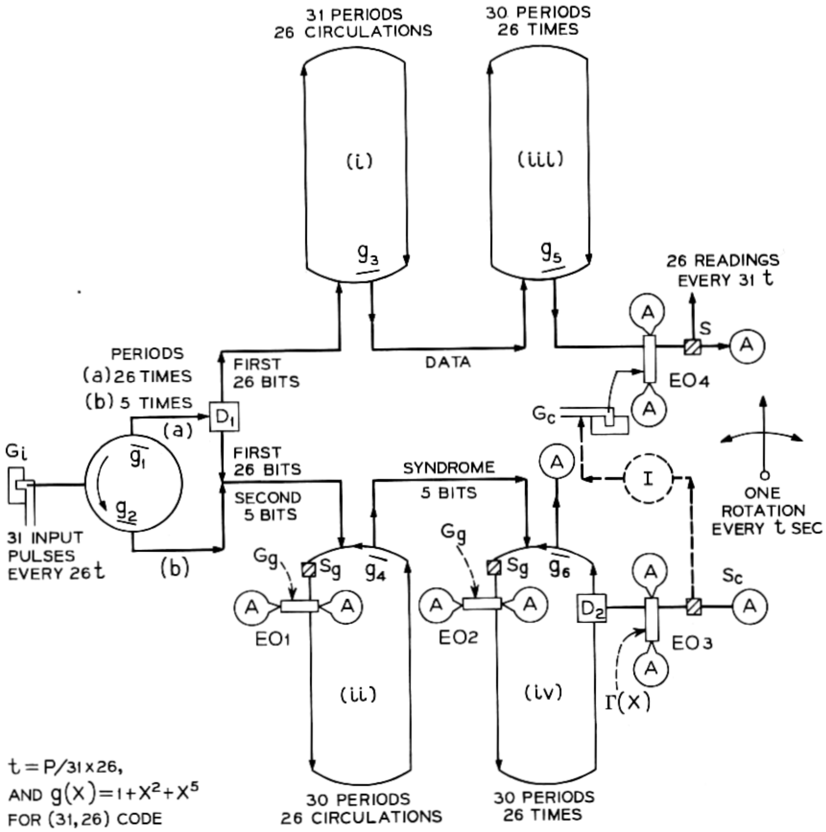


Fig. 11—Serial decoder for the (31,26) code with domains under field access propagation with $t = P/31 \times 26$.

(PART D)

V. DISCUSSION OF MAGNETIC DOMAIN ENCODERS AND DECODERS

Magnetic domain technology offers inexpensive storage but requires time for every operation (generation, propagation, exclusive-or operation, sensing, annihilation, etc.,) in contrast to the instantaneous operation of semiconductor circuitry. Such a constraint induces the variations in the design of encoders and decoders from conventional devices in semiconductor technology.

The basic functions utilized in the domain devices in this paper are: controlled generation of information, propagation, storage, duplication, gating of bubble streams, exclusive-or operation, sensing, and annihilation. Most of these functions have been successfully accomplished within the Bell System. Some of the functions are discussed in Part A and the others are reported in Refs. 8, 9, and 18. Serial arrangements of these encoders and decoders utilize fewer exclusive-or operations than the nonserial arrangements.

The packing density (which ultimately influences the active chip area in the devices) depends upon the nature of the uniaxial magnetic material chosen. Typical orthoferrites[19] ($YbFeO_3$, $YFeO_3$, and $Sm_{0.55}Tb_{0.45}FEO_3$) can hold bubbles of 40 to 50 micron diameters at 200 micron spacing yielding about 1.6×10^5 bits per square inch. Typical garnets ($Er_2Tb_1Al_{1.1}Fe_{3.9}O_{12}$ and $Gd_{2.3}Tb_{0.7}Fe_5O_{12}$) can support bubbles of 4 to 8 micron diameters at 25 micron spacing yielding about 10^6 bits per square inch. Newer epitaxial garnet films have yielded up to 1.6×10^6 bits per square inch of storage.[20] The hexagonal ferrites ($PbAl_4Fe_8O_{19}$) support bubbles of 4 to 8 microns in diameter.

The domain velocity (which ultimately influences the speed of devices) depends on the field difference across the bubble diameter and the magnetic material used. A nominal value of 20 Oe can be generated in field access propagation with a T-bar type of overlay. Orthoferrites require the lowest time to move a bubble from one position to the next data position approximately four diameters away, thus yielding a data rate of about one megacycle of 20 Oe field difference. The highest rate achieved is about three megacycles. Some of the earlier garnets have lower mobilities and a data rate of 140 kHz has been achieved with field access propagation. Some of the newer garnet films have yielded data rates of up to one megacycle.[20] Hexagonal orthoferrites have the lowest mobilities and are suitable for 10 to 60 kHz application. The data rates thus far attained in orthoferrites and garnets are sufficient to construct encoders and decoders at normal data transmission rates. For instance a transmission rate of 4800 bits per second would demand a data rate of about 125 kHz.

One of the differences between the conventional semiconductor devices and the serial type of bubble devices is the ease of converting one generator polynomial to another generator of the same degree without changing the control or propagating circuitry. If it is desired to change the generator, then it is necessary only to change the sequence of bubbles injected by G_g in Fig. 7 for the encoder and Fig. 11 for the decoder together with the generator $\Gamma(X)$, without altering the rest of the circuitry. Further, the embodiments presented indicate that the serial encoders and decoders with field access propagation yield flexible designs for block codes whose block length is about thirty bits or more.

VI. CONCLUSIONS

Magnetic domains may be used to construct encoders and single-error correcting decoders for cyclic block codes. The magnetic material chosen to host the bubbles depends on the transmission rate, and the generator of the code may be changed from one polynomial to another of the same order without altering the embodiment or the control circuitry in the serial type of devices.

In the field access propagation only one clock frequency is utilized to accomplish storage, division, and synchronizing the input and the output. The same clock excites the main propagating magnetic field once during an interval calculated as $(P/n \times k)$ seconds, where P is the time required to transmit one block of data through the transmission channel, n is the total number of bits in the block, and k is the number of information bits in the block.

VII. ACKNOWLEDGMENTS

The author thanks D. D. Sullivan, A. H. Bobeck, P. I. Bonyhard, A. J. Perneski, and E. R. Berlekamp at Bell Laboratories for the discussions during the development of the possible configurations of the magnetic domain encoders and decoders.

REFERENCES

1. Bobeck, A. H., "Properties and Device Applications of Magnetic Domains in Orthoferrites," B.S.T.J., *46*, No. 8 (October 1967), pp. 1901–1925.
2. Bobeck, A. H., "Properties of Cylindrical Magnetic Domains in Orthoferrites," IEEE Trans. Magnetics, *4*, (1968), pp. 450 ff.
3. Bobeck, A. H., and Gianola, U. F., "Magnetic Domains," Science and Technology, No. 86 (1969).
4. Perneski, A. J., "Propagation of Cylindrical Magnetic Domains in Orthoferrites," IEEE Trans. Magnetics, *5*, No. 3 (September 1969), pp. 554–557.
5. Thiele, A. A., "The Theory of Circular Magnetic Domains," B.S.T.J., *48*, No. 10 (December 1969), pp. 3287–3335.
6. Thiele, A. A., "Theory of Static Stability of Cylindrical Domains in Uniaxial Platelets," J. Appl. Phys., *41*, No. 3 (March 1970), pp. 1139–1145.
7. Bobeck, A. H., Fischer, R. F., Perneski, A. J., Remeika, J. P., and Van Uitert, L. G., "Application of Orthoferrites to Domain Wall Devices," IEEE Trans. Magnetics, *5*, No. 3 (September 1969), pp. 544–553.
8. Bobeck, A. H., Fischer, R. F., and Perneski, A. J., "A New Approach to Memory and Logic Cylindrical Domain Devices," Proc. FJCC, (1969), pp. 489–498.
9. Bonyhard, P. I., Danylchuk, I., Kish, D. E., and Smith, J. L., "Application of Bubble Devices," IEEE Trans. Magnetics, *6*, No. 3 (September 1970), pp. 447–451.
10. Hamming, R. W., "Error Detecting and Error Correcting Codes," B.S.T.J., *29*, No. 1 (January 1950), pp. 147–160.
11. Bose, R. C., and Ray-Chaudhuri, D. K., "On Class of Error Correcting Binary Group Codes," Information Control, *3*, pp. 68–79.
12. Peterson, W. W., *Error-Correcting Codes*, Cambridge, Massachusetts: MIT Press, 1961.

13. Lucky, R. W., Salz, J., and Weldon, E. J., *Principles of Data Communication*, New York: McGraw–Hill Book Company, 1968.
14. Berlekamp, E. R., *Algebraic Coding Theory*, New York: McGraw-Hill Book Company, 1968.
15. Hocquenghem, A., "Codes Correcteurs d'erreurs," Chieffres, *2*, (1959), pp. 147–156.
16. Peterson, W. W., *loc. cit.*, pp. 201–202.
17. Berlekamp, E. R., *loc. cit.*, p. 123.
18. Bonyhard, P. I., private communication.
19. Bobeck, A. H., Danylchuk, I., Remeika, J. P., Van Uitert, L. G., and Walters, E. M., "Dynamic Properties of Bubble Domains," presented at the International Conference on Ferrites, July 6–9, 1970, Kyoto, Japan.
20. Fischer, R. F., private communication.