ON THE DESIGN AND CONSTRUCTED OF A FEEDBACK SHIFT-REGISTER ENCODER

E. O. Ukem
Department of Physics, University of Calabar, P.M.B. 1115, Calabar, Nigeria

(Submitted: 10 October, 2006; Accepted: 5 February, 2007)

Abstract

Information transmission in noisy channels can be achieved with vanishingly small probability of error by proper coding of the information as long as the encoding rate is less than the channel capacity. An encoder capable of cyclical shifting of data, and which can therefore be used for Bose-Chaudhuri and Hocquenghem (BCH) coding, has been designed and constructed using discrete components. It comprises basically four bistable multivibrators and an exclusive-OR device. On completion, the encoder performed cyclic code generation satisfactorily.

Keywords: *Error coding, code rate, channel capacity, generator polynomial.*

1. Introduction

In the transmission of information, be it in the digital or analog form, there is the need for encoding before the information is sent on the channel. Among other reasons, the encoding, or conversion of the message from one form to another, may be necessary in order to guard against the occurrence of error, or, in the event of error occurring, to correct it. In data transmission, information is essentially transmitted over channels that are predominantly wires. The information is therefore invariably sent over noisy channels. Environmental interference and physical defects in the communication medium can cause errors during transmission. Error coding is a method of detecting and correcting these errors to ensure that information is transferred intact from its source to its destination. It is a method of providing reliable digital data transmission and storage when the communication medium used has an unacceptable bit error rate and a low signal-to-noise ratio. Error coding is used in many digital applications such as fault tolerant computing in computer memory, magnetic and optical data storage media, satellite and deep space communications, network communications, and cellular telephone networks (Shelton, 1999). Rather than transmit digital data in a raw form, the data is encoded with extra bits at the source. The longer code word is then transmitted, and the receiver can decode it to retrieve the desired information. The extra bits transform the data into a valid code word in the coding scheme, thereby providing redundancy that, according to the coding scheme used, will allow the destination to use the decoding process to determine if the communication medium introduced errors and in some cases correct them.

The objective of this paper is to describe the feedback shift-register encoder that was designed and constructed by the author, which can be used to carry out such encoding. The encoder was designed and constructed completely using discrete components, thus affording the opportunity for greater understanding of, and intimacy with, the circuitry.

2. Theoretical Background

The basic Information Theory, the coding theorem for noisy channels, as stated by Shannon (Shannon, 1949) ensures the existence of codes for noisy channels, which can be decoded with vanishingly small probability of error as long as the code rate is less than the channel capacity. According to the Shannon Capacity Theorem, noise-induced errors in a communication channel can be decreased to any desired level without degrading the information transmission rate, by proper encoding of information, as long as the encoding rate is

less than the channel capacity (Khanna, 1999). One approach in the search for optimum codes has been the use of the structures of abstract algebra. This has resulted in the development of a considerable body of coding theory of both theoretical and practical interest (Ukem, 1975). Classes of good and efficiently decodable codes have been generated, known as algebraic codes mainly because of their algebraic structure. Some of such codes are the Bose-Chaudhuri and Hocquenghen (BCH) codes, discovered by Hocquenghen and independently by Bose and Chaudhuri (Ukem, 1975). BCH codes have many advantages, which include the fact that their error-correcting capabilities can be pre-specified and that they have a relatively simple decoding algorithm. In technical terms, a BCH code is a multilevel, cyclic, error-correcting, variable-length digital code used to correct multiple random error patterns (Wikipedia, 2004). A significant characteristic of cyclic codes is that when any code word is rotated left or right by any number of digits, the resulting string is still a word in the code space. In other words, these are codes such that the code vectors are simple lateral shifts of one another (Schwartz, 1980). This property makes coding and decoding very easy and efficient to implement by using shift registers (Shelton, 1999).

Cyclic codes are describable in polynomial form (Schwartz, 1980). The code word, c = (c_1, c_2, ..., c_n) may be expressed as the (n-1) degree polynomial

$$c(x) = c_1x^{n-1} + c_2x^{n-2} + \ldots + c_{n-1}x + c_n \quad (1)$$

Each power of x represents a one-bit shift in time. The highest-order coefficient c_1 in the polynomial represents the first bit of the code word, while the last coefficient c_n represents the last bit of the code word. Successive shifts to generate other code words are then repeated by the operation x c(x) mod(x^n + 1). Thus, shifting once gives

$$x\,c(x) \bmod (x^n + 1) = c_2x^{n-1} + c_3x^{n-2} + \ldots + c_{n-1}x^2 + c_nx + c_1 \quad (2)$$

Shifting a second time gives

$$x^2\,c(x) \bmod (x^n + 1) = c_3x^{n-1} + c_4x^{n-2} + \ldots + c_nx^2 + c_1x + c_2 \quad (3)$$

Each cyclic code is derivable from a generator matrix g(x). A code word c(x), in polynomial form, may be written in the form

$$c(x) = a(x)\,g(x) \quad (4)$$

The generator polynomial g(x) for an (n, k) cyclic code is a divisor of x^n -1, where n is the total number of bits in the code word and k is the number of information bits, implying that the number of parity check bits is r = (n-k) (Schwartz, 1980).

Cyclic codes lend themselves readily to generation directly from the generator polynomial g(x). Shift register implementation can be used to carry out the code generation. The code word polynomial c(x) must be divisible by the generator polynomial g(x).

The encoder for BCH codes is effectively a dividing circuit. In order to effect coding, it divides the information polynomial by the generator polynomial and if there is a remainder, it adds this remainder to the information sequence to make it a code word. A code word must always be an exact multiple of the generator polynomial, thus the result of the division determines exactly what must be added to the information polynomial to make it an exact multiple of the generator polynomial. If the information polynomial is already a multiple of the generator polynomial the remainder from the division would be zero, and the check digits 0000 (as the case may be) would be added to the information sequence.

Considering a data sequence (or information sequence) d_1, d_2, ..., d_{k-1}, d_k (k information bits), it can be written in polynomial form as

$$d(x) = d_1x^{k-1} + d_2x^{k-2} + \ldots + d_{k-1}x + d_k \quad (5)$$

with degree (k-1) or less. The operation $x^{n-k}d(x)$ then generates a polynomial of degree (n-1) or less. Dividing $x^{n-k}d(x)$ by the generator polynomial g(x) of degree (n-k) results in a polynomial q(x) of degree (k-1) or less and a remainder polynomial r(x) thus

$$\frac{x^{n-k}d(x)}{g(x)} = q(x) + \frac{r(x)}{g(x)} \quad (6)$$

Since r(x) + r(x) = 0 under modulo-2 addition, it is apparent that the (n-1) degree polynomial $x^{n-k}d(x)$ + r(x) is divisible by g(x) and must therefore be a code word. Thus

$$c(x) = a(x)\,g(x) = x^{n-k}d(x) + r(x) \quad (7)$$

But $x^{n-k}d(x)$ corresponds to a simple left shift by (n-k) units of data bits. Hence the remainder r(x) must represent the parity bits (Schwartz, 1980). Specifically,

$$r(x) = \text{rem}\ \frac{x^{n-k}d(x)}{g(x)} \quad (8)$$

where "rem" denotes remainder.

The polynomial representation of cyclic codes and the calculation of the parity check bit remainder polynomial r(x) by dividing the left-shifted d(x) by the generator polynomial g(x) suggest various ways of implementing the parity check-bit calculation. One of such schemes uses r = (n-k) shift registers, shown in Fig.1. The shift register elements are designated by the 1-bit delay symbol D. As the k data bits are moving through the encoder, they are also being shifted out onto the output line, since they form the first k bits of the n-bit code word. The data bits continue moving through the shift registers until the last (i.e. the k^{th}) data bit clears the last register (n-k register). The mod-2 addition units are exclusive-OR devices. The gain controls $g_{n-k-1}, g_{n-k-2}, \ldots, g_1$ are either present (i.e. 1) or absent (i.e. 0), depending upon whether the corresponding coefficients in the g(x) polynomial given by $g(x) = x^{n-k} + g_1x^{n-k-1} + \ldots + g_{n-k-1}x + 1$ are 1 or 0 (Schwartz, 1980). At the time the last data bit clears the last register, the r = n-k registers contain the parity-check bits, which are then shifted out one at a time onto the output line, with the information source switched off. In effect, multiplication through the feedback elements provides the division called for in the calculation of the remainder polynomial.

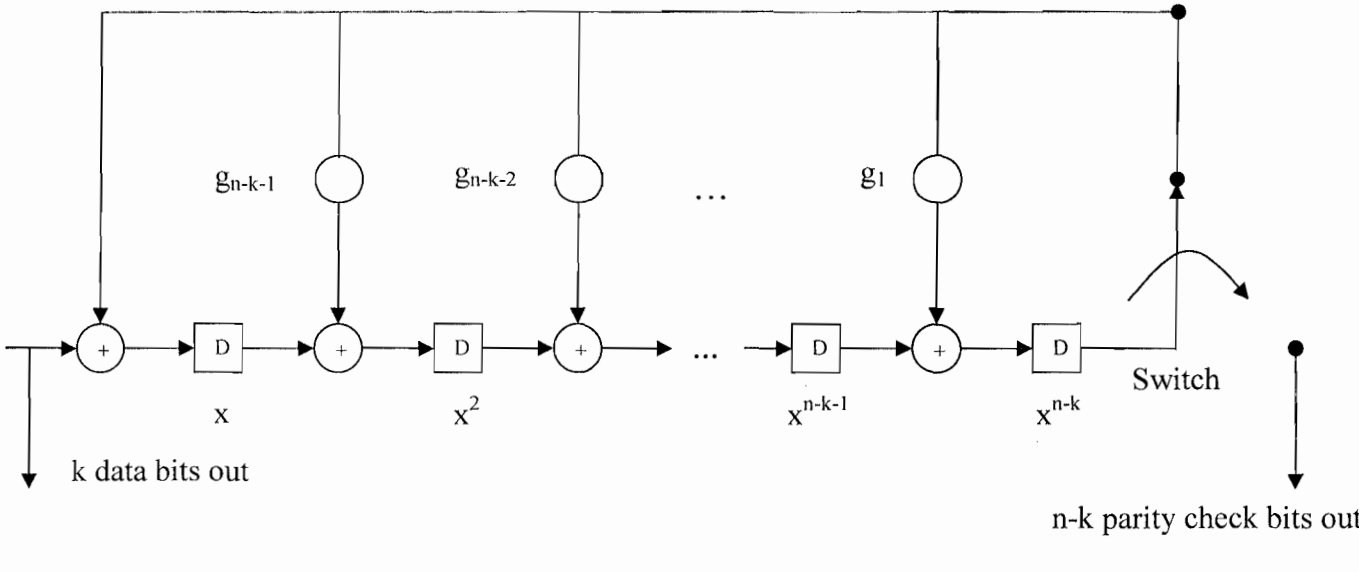


Fig.1: Cyclic code encoder, r = n – k registers (after Schwartz, 1980)

3. Materials and Methods

The class of codes selected for this work was the (15, 11) codes. That is, codes with a total block size of 15 bits, with 11 information bits and (15 – 11) = 4 parity check bits. The generator polynomial for this class of codes is $g(x) = x^4 + x + 1$ (Schwartz, 1980). From the general form of the generator polynomial

$$g(x) = x^{n-k} + g_1x^{n-k-1} + \ldots + g_{n-k-1}x + 1 \quad (9)$$

when n = 15 and k = 11 we obtain

$$g(x) = x^4 + g_1x^3 + g_2x^2 + g_3x + 1 \quad (10)$$

and, comparing this with the generator polynomial for the (15, 11) codes it is seen that $g_1 = g_2 = 0$ and $g_3 = 1$. The corresponding resulting encoder is thus as shown in Fig. 2.

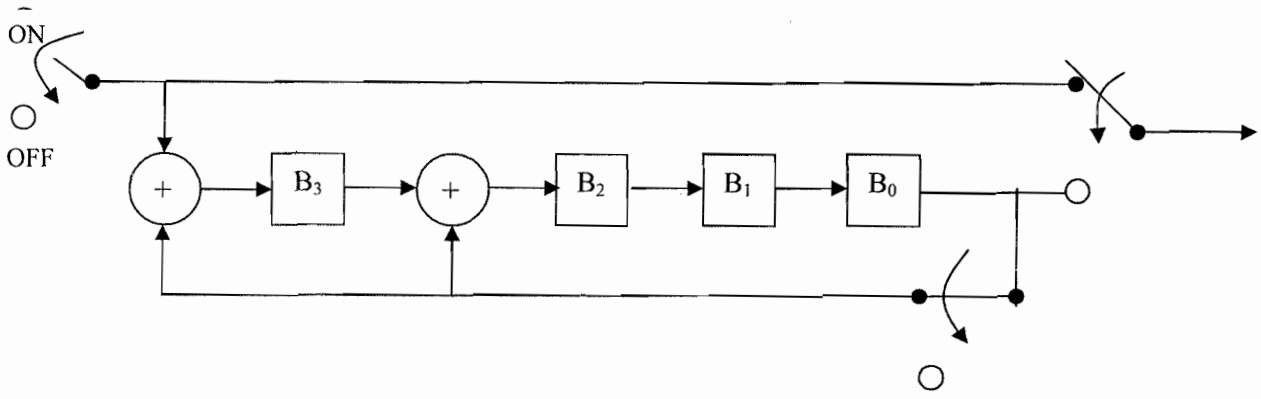
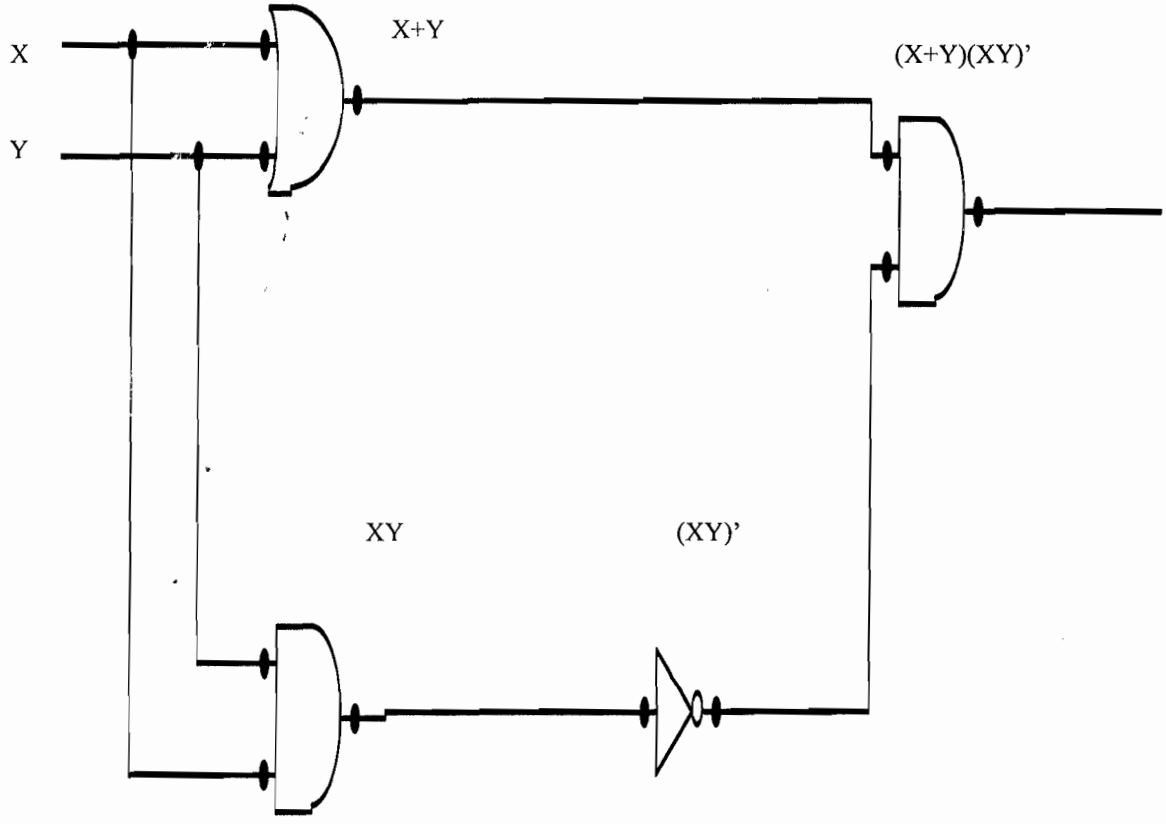


Fig. 2: Encoder for (15, 11) code. $g(x) = x^4 + x + 1$

The encoder was designed entirely with discrete components. The storage elements (registers) used were bistable multivibrators (BMV's) (or flip-flops) while the clock source was an astable multivibrator (AMV). The exclusive-OR (XOR) gate (or half-adder) circuit used was the type made of one OR-gate, two AND-gates, and an inverter, as shown in the schematic diagram of Fig. 3(a). The corresponding circuit diagram of the XOR is shown in Fig. 3(b).



(a)

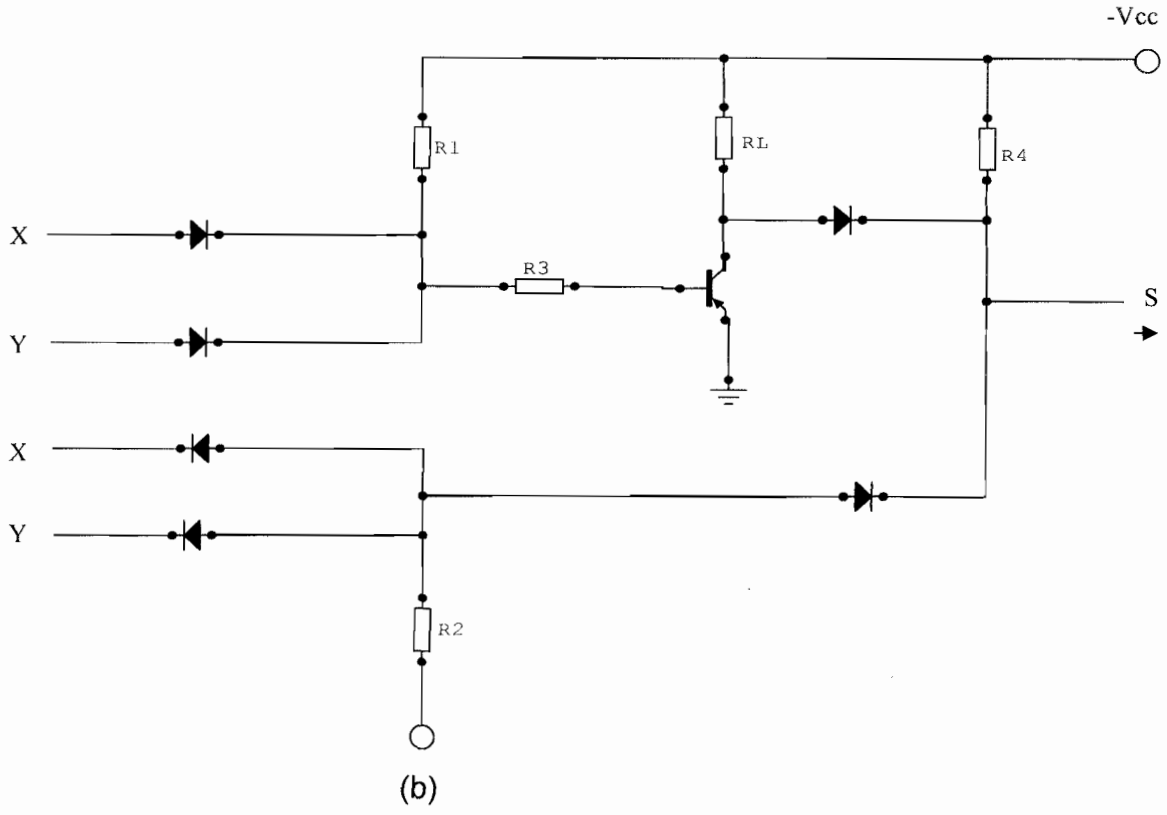


Fig. 3: Half-Adder – (a) Logic Diagram (b) Circuit Diagram

The various stages of the encoder were separately designed. From appropriate calculations (Ukem, 1975) the values of the various components were determined. In all, there were four BMV's, one XOR circuit, and one AMV. Components used in the entire encoder were:

Transistors (AC125) - 13
Resistors (various values) - 46
Capacitors (various types) - 24
Diodes (OA7 – 6No; AA119 – 8No.) -14

The encoder was assembled in accordance with the block diagram in Fig. 4. The outputs of the first and last BMV's were fed directly into the half-adder, while all other inter-stage couplings were done through electrolytic capacitors and diodes. Triggering was arranged to occur simultaneously in all the stages.

For the purpose of testing the encoder, two dual-beam oscilloscopes and an external power supply unit were employed. The dual-beam oscilloscopes were used to make it possible to display all the four outputs of the four BMV's for the purpose of comparison. First, the encoder circuit was switched on without the triggering pulses from the AMV, and the number in the registers was noted. Then the AMV pulses were switched on and the output of the four BMV's observed closely.

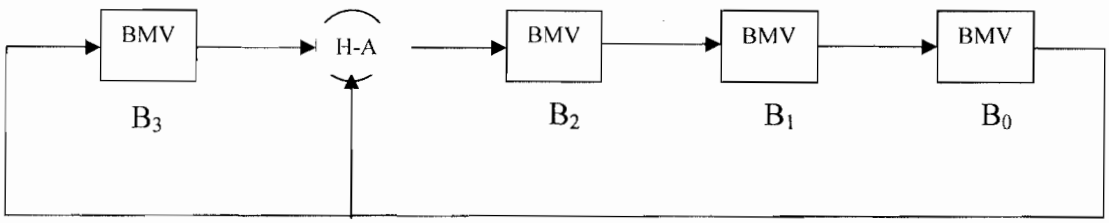


Fig. 4: Block diagram of shift register encoder

4. Results and Discussion

With the two dual-beam oscilloscopes it was possible to observe the four outputs of the BMV's (registers). The operation of the encoder was observed to be such that, if at one instant the digits held in the stages B_0, B_1, B_2, and B_3 were, respectively, 0, 0, 1, 0, the arrival of a clock pulse shifted the content of B_2 to B_1, of B_1 to B_0, and of B_0 to B_3 and the half-adder. The content of B_3 also went to the half-adder, where modulo-2 addition was carried out on the two inputs and the result stored in B_2. Considering the initial values 0, 0, 1, 0, after the first clock pulse the content of B_3 was 0, that of B_2 was 0 (i.e. 0 (+) 0 = 0), B_1 was 1 and B_0, 0. ((+) is used here to indicate modulo-2 addition). After the second pulse the stored digits were

B_3 - 0
B_2 - 0 (i.e. 0 (+) 0 = 0)
B_1 - 0
B_0 - 1

After the third pulse the contents were

B_3 - 1
B_2 - 1 (i.e. 0 (+) 1 = 1)
B_1 - 0
B_0 - 0

Table 1 shows the progression of bit patterns as the pulses were applied. From the table it is observed that each pulse produces a cyclic shift of the bits, and the sequence repeats itself after 15 pulses ($15 = 2^4 - 1$, where 4 is the number of storage elements – the BMV's)

Table 1: Bit pattern progression on application of pulses to the encoder

B_3	B_2	B_1	B_0
0	0	1	0
0	0	0	1
1	1	0	0
0	1	1	0
0	0	1	1
1	1	0	1
1	0	1	0
0	1	0	1
1	1	1	0
0	1	1	1
1	1	1	1
1	0	1	1
1	0	0	1
1	0	0	0
0	1	0	0
0	0	1	0
0	1	0	0
0	0	1	0
0	0	0	1

The encoder was found to perform satisfactorily. It was designed and built with relatively inexpensive components, and so is in itself relatively cheap. This is an advantage because it makes the encoder more easily affordable and hence more readily available. Another advantage of the encoder is that, being designed and built with discrete components, it makes for greater intimacy with the circuitry. More details of the circuit design are available than would be if integrated circuits (IC's) were employed (Ukem, 1975), and this creates an atmosphere for greater understanding of the circuit. Yet another advantage is that low power transistors were used in the encoder. This has the effect of keeping operational costs low. The major disadvantage of the circuit, however, is the size. The encoder is rather bulky due to the same fact that discrete components were used in its construction. A further setback would be the speed of operation. Relatively inexpensive, readily available audio frequency transistors were used in place of more expensive switching transistors. Although the switching operation of the transistor used (AC125) is generally satisfactory, it cannot give high frequency switching as do actual switching transistors.

The encoder in this work, due to its simplicity of design and the selection of components for its construction (low-cost discrete components), is considered to be more suitable for experimental and investigative work than live application in a communications system. It is therefore expected to be reasonably useful as training equipment or tool, to assist in the investigation of the generation of cyclic codes. More advanced error protection schemes are more complicated and are being implemented in software. However, this adds to complexity and may be less reliable than hardware encoders because software is less mature and more difficult to verify its correctness (Lin and Costello, 2004).

5. Conclusion

The encoder was designed and built, and was demonstrated to be capable of shifting, cyclically, the digits held in the registers. The device was not actually connected to an information source, so parity check bits were not generated. (To do that, an additional XOR gate would be required, and the set-up

would be as in Fig.2). The significance of the work lies in the fact that it has demonstrated a relatively inexpensive encoder that can be used for further investigation of information coding for error detection and correction.

Acknowledgements

The author wishes to acknowledge Prof. B. G. Bajoga for his valuable assistance, and the Department of Electrical Engineering, Ahmadu Bello University, Zaria, Nigeria, for providing the facilities for the study. Very special thanks are extended to Dr. M. U. Onuu of the Department of Physics, University of Calabar, Calabar, Nigeria, for his interest and encouragement, and immeasurable assistance in areas too numerous to be enumerated.

References

Khanna, V K. (1999): Digital Signal Processing, Telecommunications and Multimedia Technology, S. Chand & Company Ltd., New Delhi.

Lin, S and Costello, D. (2004). Error Control Coding: Fundamentals and Applications. Prentice-Hall, Englewood Cliffs, NJ.

Peterson, W. W. and Weldon, E. J., Jr. (1972): Error-Correcting Codes, MIT Press, Cambridge, Mass., 2nd ed.

Schwartz, Mischa (1980): Information Transmission, Modulation, and Noise, 3rd Ed., McGraw-Hill Kogakusha Ltd., Tokyo.

Shannon, C. E. (1949): "Communication in the Presence of Noise", Proc. IRE Vol. 37 pp 10 – 21.

Shelton, Charles P. (1999): Coding for Error Detection and Correction, 18-84b Dependable Embedded Systems, Carnegie Mellon University.

Ukem, E. O. (1975): The Design and Construction of a Shift Register Encoder, Final Year Project, Department of Electrical Engineering, Ahmadu Bello University, Zaria, Nigeria.

Wikipedia: http://en.wikipedia.org/wiki/BCH_code