Meggitt Decoder – An FPGA Implementation

D. Avinash and Dr.P. Srihari

Abstract--- *This paper presents the FPGA implementation of Meggitt decoder used for decoding the cyclic codes. However, whether or not the decoder will be practical will depend on the complexity of the combinational logic circuits in the error detector. Special cases of codes have been developed that leads to simpler logic circuits in the decoder. However, this is accomplished at the expense of lowering the efficiency of the code for a given block size. Meggitt decoder avoids the need for a syndrome table by computing the error syndromes of all correctable error patterns from a small number of error syndromes. In the case of single error correcting code, all the syndromes can be determined from any one error syndrome. In place of syndrome table, this Meggitt decoder has a detection circuit whose input is the error syndrome and whose output is 1 or 0. This work presents generalized Meggitt decoder for (n, k) cyclic code and implemented (7, 4) cyclic code by using SPARTRAN 3 FPGA.*

Keywords--- *Cyclic Codes, Encoder, FPGA, Meggitt Decoder, Syndrome*

I. INTRODUCTION

1.1 Error Control Coding Schemes

THE codes referred to in the channel coding theorem do not prevent the occurrence of errors but rather allow their presence to be detected and corrected. As such, the codes are known as error-detecting and error correcting codes or error-control codes. Error control codes are classified into two categories: block codes and convolutional codes.

1.2 Channel coding Schemes

1.2.1 Block Codes

A block code is a set of words that has a well-defined mathematical property or structure, and where each word is a sequence of a fixed number of bits. The words belonging to a block code are called codewords. Note that m-bits is referred to as an m-bit word.

A code word consists of information bits that carry information and parity-check bits(also referred to as parity bits or check bits) that carry no information. A code whose code words have k information bits and r parity bits, has n-bit code words where n=k+r. Such a code referred to as an (n,k) block code where n and k are respectively the block length and information length of a code. A codeword whose information bits are kept together, so they are readily identifiable, is said to be in a systematic form or to be systematic, otherwise the codeword is referred to as non-systematic.

1.2.2 Cyclic Codes

Cyclic codes are also Block codes. An (n,k) linear code C is cyclic if every cyclic shift of a codeword in C is also a codeword in C.

If $c_0 \; c_1 \; c_2 \; \ldots \; c_{n-2} \; c_{n-1}$ is a codeword, then

$c_{n-1} \; c_0 \; c_1 \; \ldots \; c_{n-3} \; c_{n-2}$

$c_{n-2} \; c_{n-1} \; c_0 \; \ldots \; c_{n-4} \; c_{n-3}$

$\vdots \quad \vdots \quad \vdots \quad \quad \vdots \quad \vdots$

$c_1 \; c_2 \; c_3 \; \ldots \; c_{n-1} \; c_0$ are all codewords.

Cyclic structure makes the encoding and syndrome computation very easy. An (n,k) linear block code C is called a cyclic code if it has the following property:

If an n-tuple $V=(v_0,v_1,v_2,\ldots..,v_{n-1})$ is a code vector of C,then the n-tuple $V^{(1)}=(v_{n-1},v_0,v_1,\ldots..v_{n-2})$ obtained by shifting V cyclically one place to the right is also a code vector of C .

As cyclic codes are linear, encoding and decoding can be achieved in the same way as for linear codes, namely using matrices. However, with cyclic codes, a polynomial representation of codewords is often more suitable and encoding and decoding rules can be formulated in terms of polynomial algebra.

1.2.3 Polynomials and Generator Polynomials

An n-bit word $(a_{n-1},a_{n-2},\ldots.,a_2,a_1,a_0)$ can be represented by the polynomial,

$$p(x) = a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\ldots\ldots+a_2x^2+a_1x+a_0$$

where the n components of the word $a_{n-1},a_{n-2},\ldots..a_2,a_1,a_0$ form the polynomial coefficients. Note that the highest possible power of x in the polynomial representation of an n-bit word is n-1. In an (n,k) cyclic code there exists a unique generator polynomial from which all the codeword polynomials can be generated. The generator polynomial is unique within the code and none of the other codeword polynomials are able to generate the complete set of codeword polynomials. For an (n,k) binary cyclic code the generator polynomial has the form,

$$g(x) = g_{n-k}x^{n-k}+g_{n-k-1}x^{n-k-1}+\ldots\ldots..+g_2x^2+g_1x+g_0$$

where the coefficients $g_{n-k-1},g_{n-k-2},\ldots\ldots..g_2,g_1$ are 0 or 1.

The coefficients of first and last terms are always equal to 1, and so

$$g_{n-k} = 1$$
$$g_0 = 1$$

The x^{n-k} term is always present in the generator polynomial and there are no higher order terms, therefore the degree of the generator polynomial g(x) of an (n,k) cyclic code is n-k. the

D. Avinash, Department of ECE, VNR VJIET, Hyderabad. E-mail: avinash.mtech2@gmail.com

Dr.P. SRIHARI, Department of EIE, Gitam University, Hyderabad, E-mail: mail2pshari@yahoo.com

ISBN 978-93-82338-26-0 | © 2012 Bonfring

generator polynomial g(x) of an (n, k) cyclic code is always a polynomial factor of the polynomial x^n-1,(or x^n+1). This means that any factor of x^n-1 of degree n-k is a possible generator of an (n, k) cyclic code.

II. GENERALIZED MEGGITT DECODER IMPLEMENTATION FOR (N, K) CODES

2.1 Block Diagram of Meggitt deCoder for (n, k) Cyclic Code

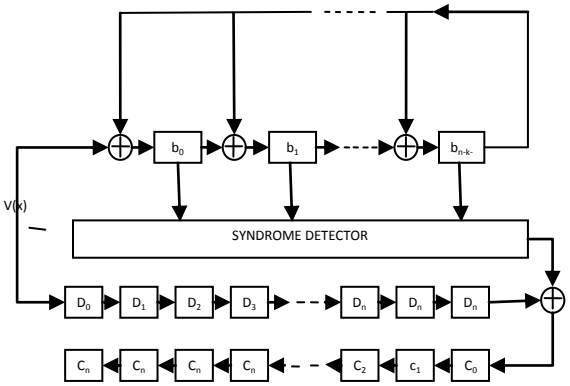

Figure (2): A Decoder Using a Syndrome Table for Error Correction

2.2 Operation of General form of a Decoder (Meggitt Decoder) for Cyclic Codes

The decoding operation can be implemented using the system shown in fig (2).

The error correction procedure consists of the following steps:

1. *Step1*: The received vector is shifted into the buffer register and the syndrome register.
2. *Step2*: After the syndrome for the received vector is calculated and is placed in the syndrome register, the contents of the syndrome register are read into the detector. The detector is a combinatorial logic circuit design to output a 1 if and only if the syndrome in the syndrome register corresponds to a correctable error pattern with an error at the highest order position x^{n-1}. That is, if the decoder output is 1, the received digit at the right-most stage of the buffer register is assumed to be erroneous and hence is corrected. If the detector output is 0, then the received digit at the right-most stage of the buffer register is assumed to be correct. Thus the detector output is the estimated error value for the digits coming out of the buffer register.
3. *Step3* : The first received digit is shifted out of the buffer and at the same time the syndrome register is shifted right at once. If the first received digit is in error, the detector output will be 1, which is used for correcting the first received digit. The output of the detector is also fed to the syndrome register to modify the syndrome. This results in a new syndrome corresponding to the altered received vector shifted to the right by one place.
4. *Step4*: The new syndrome is now used to check whether or not the second received digit, now at the right-most stage of the buffer register, is an erroneous digit. If so, it is corrected, a new syndrome is calculated as in step3 and procedure is repeated.
5. *Step5*: The decoder operates on the received vector digit by digit until the entire received vector is shifted out of buffer. At the end of the decoding operation (that is, after the received vector is shifted out of the buffer), errors will have been corrected if they correspond to an error pattern built into the detector and the syndrome will contain all zeros. If the syndrome register does not contain all zeros, then an uncorrectable error pattern has been detected. The decoder shown in fig (2) can be used for decoding any cyclic code. The present work aims at FPGA implementation of Meggitt Decoder.

III. MEGGITT DECODER DESIGN IMPLEMENTATION FOR (7, 4) CODE

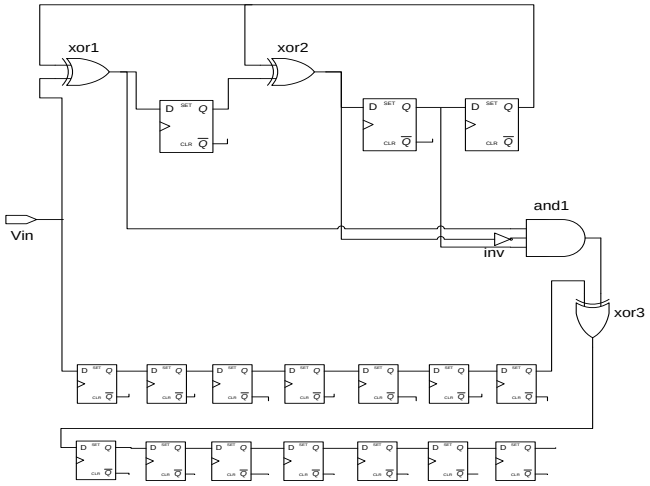

Figure (3): Schematic Diagram of Meggit Decoder (7,4) Code

The above fig (3) shows a Meggitt decoder for the (7,4) code with generator polynomial $g(x) = x^3+x+1$. Consider again the codeword $c(x) = x^6+x^4+x+1$ incurring the error x^3 so giving $v(x) = x^6+x^4+x^3+x+1$ as the word to be decoded. The input to the decoder is $(v_6,v_5,v_4,v_3,v_2,v_1,v_0) = (1011011)$ where v_3 is in error. The detector is arranged to detect the error syndrome of the highest-order bit, so that here we have $s_d(x) = R_{g(x)}[x^6]$ and given the generator polynomial $g(x) = x^3+x+1$ this gives $s_d(x) = x^2+1$. Hence the detector needs to detect when $b_0=1, b_1=0$ and $b_2=1$. This is achieved by feeding b_0, b_1 and b_2 into a 3-input AND gate with the b_1 input passing through an inverter. When $b_0=1, b_1=0$ and $b_2=1$ all 3 inputs to the AND gate are 1 and so the gate's output $s_d=1$. At all other times the gate's output is 0.

IV. SIMULATION RESULTS AND DISCUSSIONS

4.1 Error Free Output of the Meggitt Decoder for (7, 4) Cyclic Code

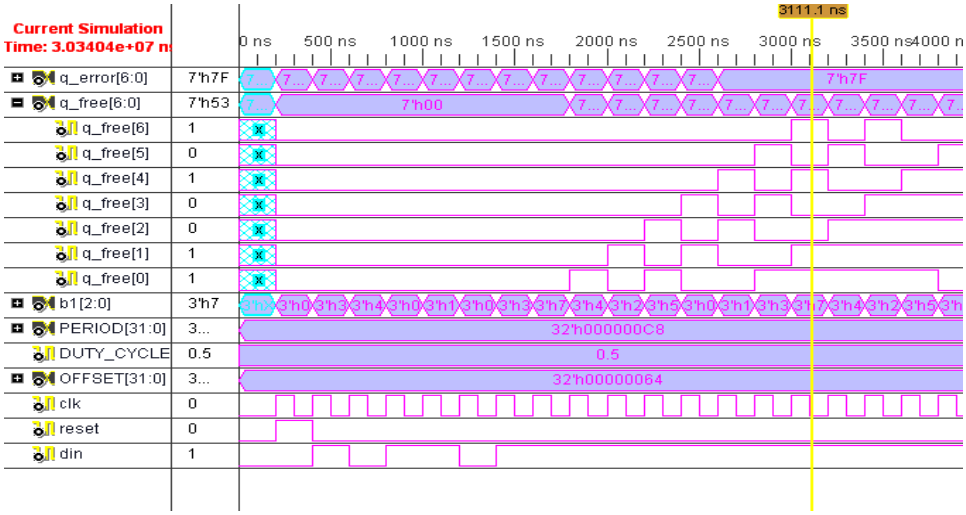

Figure (4): Simulation Results of (7,4) Code Decoder

Fig (4) shows the step-by-step operation of (7,4) code decoder output in which the error is placed in the received codeword i.e., (1011011) and after seven cyclic shifts the error free codeword is (1010011). Here fourteen shifts are required for decoding, of which the first seven are needed to obtain the error syndrome of v(x) and the last seven for error correction.

4.2 Syndrome Design of the Meggitt Decoder for (7, 4) Cyclic Code

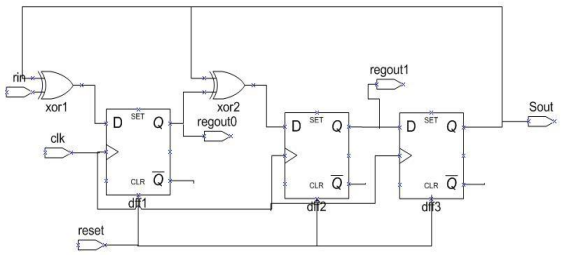

Figure (5): Syndrome Design for (7, 4) Cyclic Code

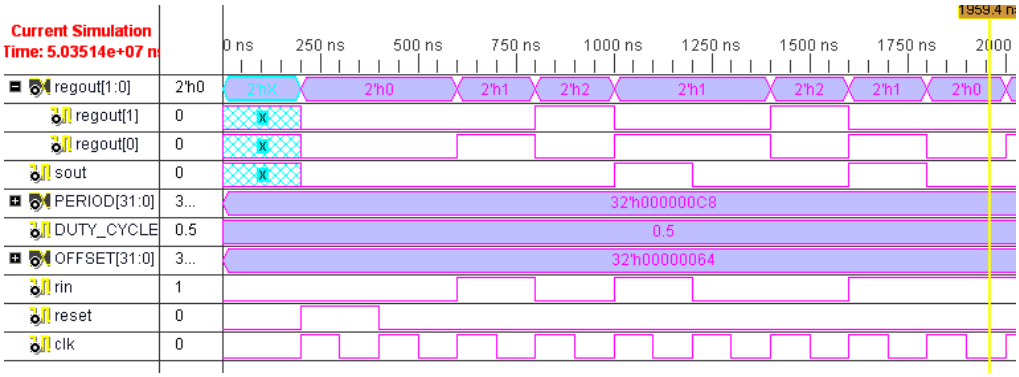

Figure (6): Simulation Results for (7,4) Code Syndrome Detector

The decoder's step-by-step operation is shown in Table 2 in which fourteen shifts are required for decoding, of which the first seven are needed to obtain the error syndrome of v(x) and the last seven for error correction respectively. Here we placed an error input as 'x^3' term and after error correction the received codeword is obtained at fourteenth shift respectively.

In the above Fig (5), the generalized syndrome detector of (7, 4) cyclic code consists of n-k flip flops i.e., three flip flops to design syndrome detector. The received code 1100101 checked by the syndrome detector and checks whether the codeword is with error or without error respectively.

4.3 Error Free Syndrome Output of the Meggitt Decoder for (7, 4) Cyclic Code

Fig (6) shows the syndrome operation for received codeword 1100101 for (7, 4) cyclic code. Its operation shows in below Table 1 respectively.

ISBN 978-93-82338-26-0 | © 2012 Bonfring

4.4 Table of the Meggitt Decoder Error Free Output for (7, 4) Cyclic Code

Table 2: Meggitt Decoding for (7, 4) Cyclic for the Codeword 1100101.

Input $v(x) = x^6 + x^4 + x^3 + x + 1$

Shift	b_{in}	b_0	b_1	b_2	b_f	e_0	e_1	e_2	e_3	e_4	e_5	e_6	d_0	d_1	d_2	d_3	d_4	d_5	d_6	c_0	c_1	c_2	c_3	c_4	c_5	c_6
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0
7	1	1	1	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0
8	—	—	—	—	—	0	0	0	0	1	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0	0
9	—	—	—	—	—	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0
10	—	—	—	—	—	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	1	0	0	0	0
11	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0
12	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0
13	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0
14	—	—	—	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1

V. FPGA OUTPUT OF THE MEGGITT DECODER FOR (7, 4) CYCLIC CODE

Figure 7: FPGA Output of the Meggitt Decoder for (7, 4) Cyclic Code.

In the above Fig 7, the output is shown in the LED's such that LED's ON shows the parity check bits for the given message bits 0101 respectively. Finally the output of Meggitt decoder for given message bits 0101 is 1100101 respectively.

Hence FPGA implementation is shown for Meggitt decoder for (7, 4) cyclic code. The coding is written in Verilog HDL programming language.

REFERENCES

[1] K. Sam Shanmugam, "Digital and Analog communication systems", John Wiley & Sons publications, 1st edition, 2008.

[2] William Cary Huffman ,"Fundamentals of error-correcting codes ", Cambridge university press,2nd edition,2010.

[3] George C.Clark,Jr.,and J.Bibb Cain,"Error correction coding for Digital communications", Springer publications,2nd edition,1988.

[4] Krzyszt of Wesolowski, "Introduction to Digital communication systems",John Wiley & Sons publications,3rd edition,2009.

[5] J.E. Meggitt, "Error Correcting Codes and Their Implementation for Data Transmission Systems," IRE Trans. Information Theory, vol. 7, pp. 232-244,1961.

[6] Wonjin Sung, Coffey, J.T., "Optimal covering polynomial sets correcting three errors for binary cyclic codes", Information Theory, IEEE Transactions on, On page(s): 985 - 991, Volume: 48 Issue: 4, Apr 2002.

[7] Bernard Sklar & Pabitra kumar Ray, "Digital communications", Pearson education publications, 2nd edition,2009.

Avinash darise received the B.Tech degree in electronics and communication engineering from Mahatma Gandhi Institute of technology, affiliated to Jawaharlal Nehru Technological University Hyderabad, AP, India, in 2009, he is pursuing the M.Tech in VLSI System Design at VNR Vignana Jyothi Institute of Engineering & Technology, Bachupally, Hyderabad, India.

Dr. P. Sri Hari has obtained his B.E (ECE) Degree from S.R.K.R.Engineering College, Bhimavaram, affiliated to Andhra University, Visakhapatnam in 1988, M.S in Electronics and Control from BITS, Pilani in 1992, M.Tech in Digital Systems and Computer Electonics from JNTU, Hyderabad in 2001 and PhD in Electronics and Communication Engineering from JNTU, Hyderabad in 2007. He started his career as Lecturer at S.V.H College of Engineering, Machilipatnam and served for 9 years. Then he worked as Lecturer, Senior Lecturer and Associate Professor at VNTVJIET, Hyderabad. Then he was promoted as Professor and Head of the Department. Dr P. Sri Hari is a Life Member of Indian Society for Technical Education.

ISBN 978-93-82338-26-0 | © 2012 Bonfring