Industrial Engineering Journal

ISSN: 0970-2555

Volume : 53, Issue 5, May : 2024

DESIGN AND SIMULATION OF CRC ENCODER AND DECODER USING VERILOG

[1]K.Sravanthi,[2] G Naga Deepthi,[3]R.L.B.R Prasad Reddy

[1]PG Student, Dept. of ECE, SITS, Kadapa, AP, India.

[2]Assistant Professor, Dept. of ECE, SITS, Kadapa, AP, India.

[3]Associate Professor, Dept. of ECE, SITS, Kadapa, AP, India.

[1]sravanthik683@gmail.com, [2]godenadeepthi@gmail.com ,[3]rajendra.409@gmail.com

Abstract— In this paper Cyclic Redundancy Check (CRC) technique is an efficient error detection method which used to detect single and burst errors. CRC technique adds redundancy bits to the original data. The redundancy bits represent the remainder of division between the original message and the selected polynomial. At the receiver side, the received data can be recognized as valid or not. In this paper, an efficient CRC encoder and decoder circuits have been designed and implemented using Verilog HDL. Xilinx ISE 16.1 Simulator is used for circuits verification and validation for CRC (Cyclic Redundancy Checking with an input 8-bit polynomial), 5 and 8-bit input data. The results reveal that the proposed circuits are efficient in terms of hardware utilization rate.

Keywords: CRC code, error detection, Redundancy bits, VHDL language, Xilinx ISE 16.1 Simulator.

I. INTRODUCTION

Most of communication system protocols use one or more method to detect errors, so to correct it, by retransmission data or using Forward Error Correction (FEC) methods [1]. One of the most commonly method in digital communication system to detect error is Cyclic redundancy check (CRC), which used in digital network or at storage device to detect errors. CRC codes is based on cyclic error correction codes theorem which is used systematic codes to encode the original data (message) for error detection at the receiver side [2].

This method was Invented by W. Wesley Peterson, and published in 1961, which is a type of linear block codes deals with systematic error detecting code used a group of error control bits appended to the end of the message block. The error control bits represent the remainder of a division between the original message and the generator polynomial [3].

The importance of CRC method came from its burst-error detection capability and all single error with an arbitrary data (message) length. The received message pass through a sequence of operations to detect if it has an error or not. The receiver has the ability to send retransmission requests back to the data source through a feedback channel. The transmitter retransmits correct data again or other methods like hamming code are added to correct errors by the receiver directly [4].

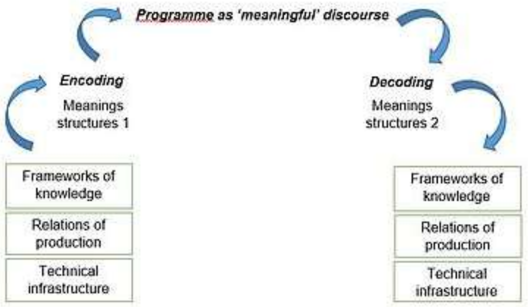

Figure – 1: Encoding and decoding of broadcast structures

II. CRC DECODER AND ENCODER

The basic idea of CRC is binary division which is different from other error detection method which is based on parity check. The CRC depends on the remainder of division at the transmitter (CRC Encoder) which it is added to original data and transmitted as shown in Figure-2.

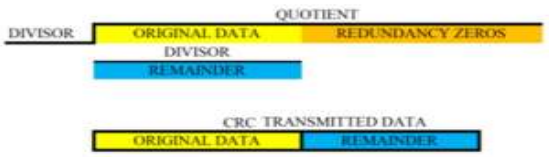

Figure – 2: CRC Encoder Procedure

There are two main points for CRC should be taken in the hardware design. The first one is that the remainder and number of redundancy zeros at encoder is equal to each other, and less than the divisor bits by one bit. As an example, if the remainder length is n-bit then the number of extra zero bits added to original data frame are n-bit long too. The number of divisor bits are n+1 then the remainder bits are appended to the original data frame (message) [5,6]. Second one is that the received data is divided by the same divisor which used at the transmitter.

At the receiver CRC Decoder) the incoming information bits (data unit) is divided by the generator polynomial, then the remainder of division is lead to knowing if data unit is correct or corrupted. If the string of bits arrives without error, the CRC Decoder checkers get a zero remainder, if it has been corrupted during transmission the division remainder is not equal to zero, as shown in Figure -3.

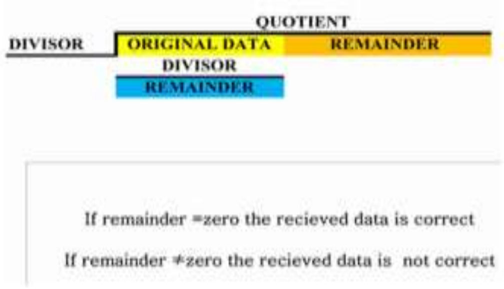
QUOTIENT
DIVISOR
ORIGINAL DATA
REMAINDER
DIVISOR
REMAINDER
If remainder =zero the recieved data is correct
If remainder ≠zero the recieved data is not correct

Figure -3: CRC decoder Procedure

III. SYSTEM DESIGN

In this project the CRC Encoder and Decoder circuits are designed with divisor polynomial X + X+X +1 which is used at Asynchronous Transfer Mode (ATM) headers. ATM headers are a type of frame synchronization using in ATM protocol and other similar protocols. The CRC-based framing was developed to improve the efficiency of a pre-standard (ATM) protocol links. This technology is using in the ATM protocols principal link, and was one of the most significant developments of StrataCom. The CRC method - Based framing re-using the header (CRC), which is present in ATM and other similar protocols to provide framing with no overhead adding on the link. In ATM, this is Known as the Header Error Control/Check (HEC) field [7, 8].

StrataCom explains the early (pre-standard) of ATM as a commercial product, (named the IPX). StrataCom's first product was T1(T1 is a time-division multiplexing) with1.544 Mbit/s-based links and it included a 5-bit CRC header, similar to ATM's 8-bit CRC header [3,9,10]. So, to design CRC encoder a random input data is supposed and the sequence procedure is explained as shown in Figure – 4 A and B and Table 1 shows the CRC encoder parameters.

Table – 3: CRC Encoder Parameters

Random header data bits	Divisor data bits X^8+X^2+X+1	CRC	The transmitted data from CRC ENCODER
10111	100000111	01100101	1011101100101
10111001	100000111	0100110	101110010100110

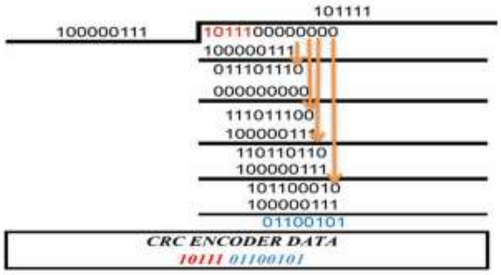

Figure – 4(A): CRC – 8 Encoder with Random 5-bit Input Data

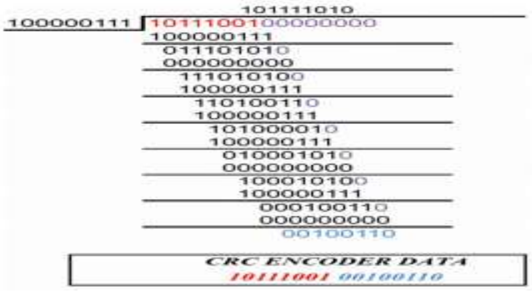

Figure – 4(B): CRC – 8 Encoder with Random 8-bit Input Data

The CRC Encoder block diagram is shown in Figure – 5 A and B, its pins desecration is explained in Table 4, the summary of CRC encoder is explained in Table 2, and the time simulation is shown in Figure – 6.A and B.

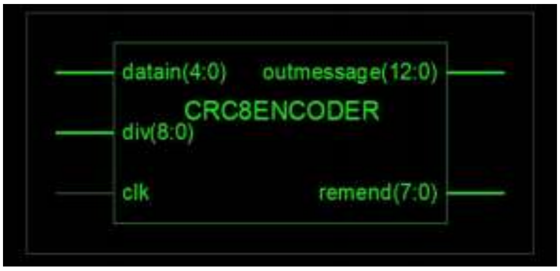

Figure – 5(A): CRC Encoder Circuit Block for 5-bit data

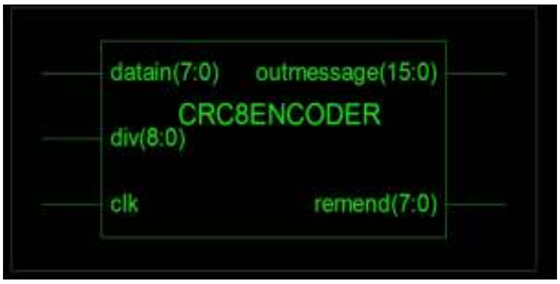

Figure – 5 (B): CRC Encoder Circuit Block for 8-bit data

Table – 2: PIN Description of CRC encoder Circuit

Signal	Direction	size	Description
CLK	Input	1	Clock signal that clocks all internal component.
DIV	Input	9	Input divisor data represented in binary bit stream..
DATAIN	Input	5 8	Input data represented in binary bit stream.
OUTMESSAGE	Output	13 16	Output data with CRC represented in binary bit.
REMEND	Output	8	Output data represents CRC remainder of binary division bit.

After data frame merged with CRC, it transmitted through transmission medium where it can be received correct or corrupted, zero or none zero division remainder. The output of the encoder for the two cases are passed through the CRC-8 decoder B. to discuss these two situations. We take the transmitted data from CRC-8 encoder and process it at (CRC - 8) decoder circuit as shown in Figure – 6 A and B, Table - 3 shows the designed CRC Decoder parameters with corrected and corrupted received data.

Table – 3: CRC Decoder Parameters

Random data bits	Divisor data bits x^8+x^2+x+1	Division Remainder	Received Data State
1011101100101	100000111	00000000	Correct
1011100100100110	100000111	00000000	Correct
1011101100100	100000111	00000001	Corrupted
1011101100100111	100000111	00001111	Corrupted

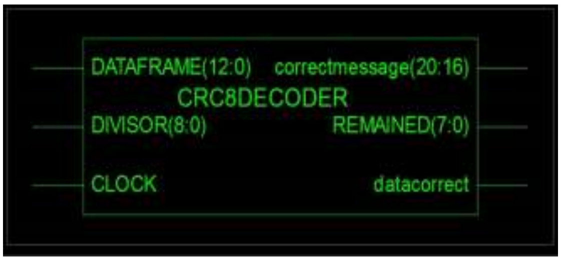

Figure – 6 (A): CRC Decoder Circuit Block With 13 Binary Bits Received Data

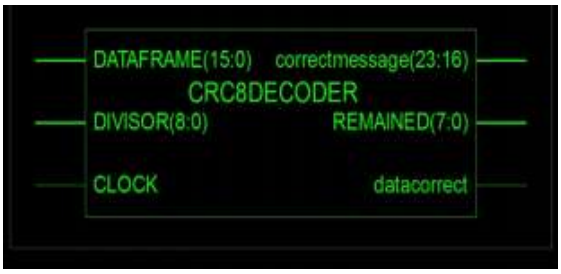

Figure – 6 (B): CRC Decoder Circuit Block With 16 Binary Bits Received Data

IV. VERIFICATION OF RESULT

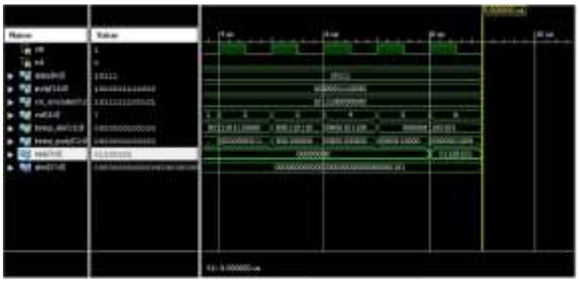

Figure – 7: CRC-8 Encoder with Random 5-bit Input Data Waveforms

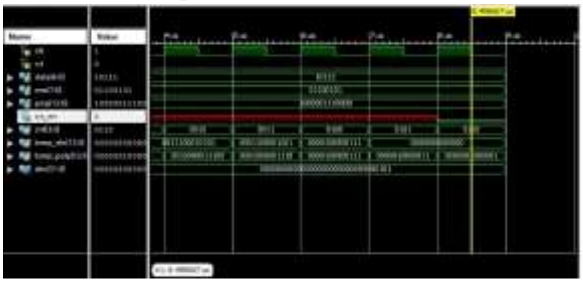

Figure – 8: CRC-8 Decoder with Random 5-bit Input Data Waveforms

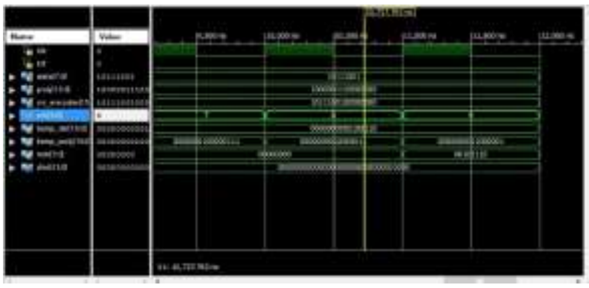

Figure – 9: CRC-8 Encoder with Random 8-bit Input Data waveforms

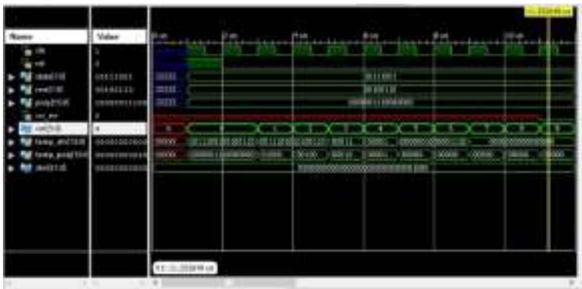

Figure – 10: CRC-8 Decoder with Random 8-bit Input Data Waveform

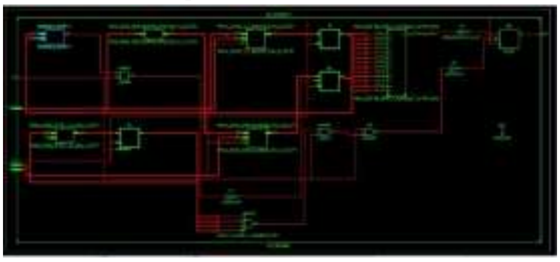

Figure –11: RTL Schematic CRC-8 Decoder with Random 5-bit Input Data

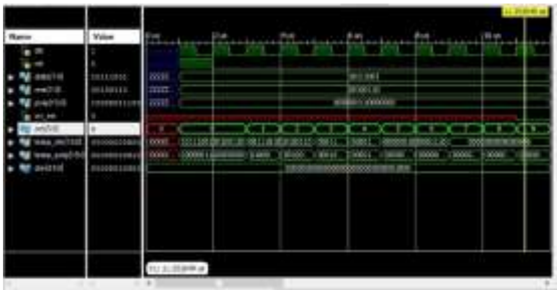

Figure - 12: RTL Schematic CRC-8 Decoder With Random 8-bit Input Data

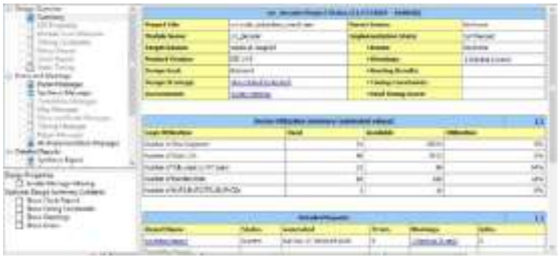

Figure – 13: RTL Schematic CRC-8 Project Summary with Random 5-bit Input Data

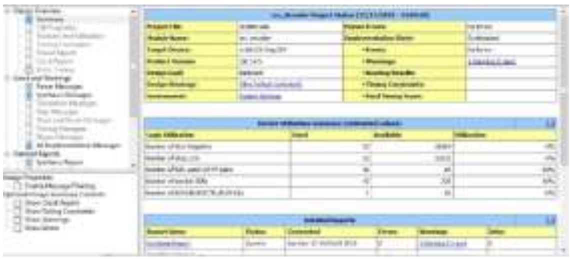

Figure -14: RTL Schematic CRC-8 Project Summary with Random 8-bit Input Data

CONCLUSION

CRC Encoder and Decoder circuits are designed and implemented using VHDL successfully. The complete process of CRC Encoding, decoding and error detection are shown in the previous section. The decoder output indicates whether the data are correct or corrupted through the output pin (data correct) , and if data correct (data correct =1) then the original message is appeared without the redundancy bits (CRC) at output pin (correct data), time simulation result is match with expected output data which explain in TABLE I and TABLE III. The proposed circuits show that low hardware usage are required which make them suitable to be integrated with any other system.

REFERENCES

[1] Forouzan, A. B., Data Communications & NetworKing (sie). Tata McGraw-Hill Education, hapter 10, p.p 265-278, 2011

[2] en.wiKipedia.org/wiKi/Cyclic_redundancy_checK 4.

[3] I. S. Satran, D. Sheinwald, “Out of Order Incremental CRCComputation”, IEEE Trans. omp., vol. 54, pp. 1178–1181,2005.

[4] Tam, S., Single error correction and double error detection.Xilinx Application Note, 2006.

[5] E. Stavinov, “A practical Parallel CRC Generation Method”, Circuit Cellary, 2010.

[6] William Stallings, Data and Computer Communications, Tenth Edition. Pearson Education, Inc., publishing as Prentice Hall,chapter 6, p.p 194, 2014.

[7] M. Divya, K. VinodalaK Shmi, G. Sumalatha, Shahbazsari K, AbhisheK Awasthi,' Design and Implementation of Encoder for (15, K) Binary BCH Code Using VHDL and eliminating the fan-out', IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), Volume 7, Issue 6 (Sep. - Oct. 2013), PP 01-06.

[8] M.Y. Rhee - “Error Correcting Coding Theory”, McGraw-Hill, Singapore, 1989.

[9] GoresKy, M. and Klapper, A.M. Fibonacci and Galois representations of feedbacK-with-carry shift registers, IEEE Transactions on Information Theory, Nov 2002, Volume: 48, On page(s): 2826 –2836.

[10] M.A.; Barbetta, L.; Neri, F.; “A Petri net simulation model of HDLC Marsan”,TENCON '89. Fourth IEEE Region 10 International Conference 22-24 Nov. 1989Page(s):240– 247Digital Object Identifier 0.1109/TENCON.1989.176933.