ICRMAT 2020 Proceedings of the International Conference on Research in Management & Technovation pp. 131–136
DOI: 10.15439/2020KM13
ISSN 2300-5963 ACSIS, Vol. 24

Implementing Cyclic Redundancy Check asError Correction Technique in HDLC

Shardeep Kaur Sooch[1], Dr Meenu Gupta[2], Dr Rakesh Kumar[3]
Chandigarh University Punjab, India
[1]*shardeep.sooch@gmail.com*, [2]*meenu.e9406@cumail.in* , [3]*rakesh77kumar@yahoo.com*

***Abstract*—Any successful communication is governed by some set of rules to manage the flow control of the transmitted data. One such protocol is High-level Data Link Control (HDLC) which is a bit-oriented protocol used for communication over the point to point or multipoint links. Residing in the data link layer (layer 2) of Open System Interconnection (OSI), this protocol transmits data in frames. HDLC can be used for detecting the errors in the data which are induced during the transmission from sender to receiver. This paper focuses on not only detecting the error but also correcting it by using Cyclic Redundancy Check (CRC). Cyclic codes are a special type of linear Block Codes in which one codeword can be cyclically shifted to obtain another codeword. The CRC generator is modulo-2 added with the data in the information frame of HDLC and the remainder is obtained. When this data is sent over any transmission channel, there are high chances of data being erroneous due to interference of unrequired signals in the channel. When data reaches the receiver end, a similar modulo-2 addition is carried to obtain another remainder. This remainder is compared with the remainder transmitted by the sender. The two compared remainders detect the location of the error bit which is corrected by flipping that specific bit. This reduces the need for Automatic Repeat Request (ARQ) mechanisms to obtain the correct information as the data can be self-corrected at the receiver end.**

***Index Terms*—HDLC, Frame Check Sequence, Cyclic redundancy Check, Xilinx, Automatic Repeat Request**

I. Introduction

WHEN any data is sent through a communication channel from the transmitter to the receiver, the data usually gets corrupted due to various reasons. The errors may be introduced in the data because of noise during data transmission, hardware failure or data read operations from memory. For the efficient and error-free transmission Error Detection and Coding technique (EDAC) is required [1]. Many codes and techniques can be implemented depending on the type of data, type of error occurring, and channel used for transmission. The basic aim for the EDAC codes is:

1. Networks must transmit data from one point to another without corrupting data with errors.
2. Error induced in the data must be detected and corrected for efficient communication.
3. Detection and correction of error are done at either the data link layer or transport layer of the Open System Interconnection (OSI) model.

Automatic Repeat Query (ARQ) is usually used as an error-control mechanism that re-transmits data in case the corrupted data is received at the receiver end. This mechanism uses the concept of acknowledgment and timeouts for better data transmissions. But for every erroneous data transmission, ARQ may not be feasible. So error correction techniques are required at the receiver end. Here CRC is being used as error detection and coding technique at transmitter and receiver. HDLC protocol, a short term coined for high-level data link control, is the most commonly used protocol for transmission as it shows high efficiency and powerful error detection. It is a bit-oriented protocol which means that it sends the information as a sequence of bits. The frames are used as a transport mechanism to transport data and contain error checking information that ensures error-free transmission. The HDLC embeds the information in the frame that allows the correction of errors and controls the data flow.

A. HDLC Frame Format

The HDLC Frame is composed of the following fields [2]:

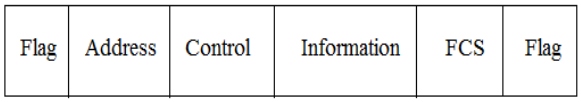

Fig. 1. HDLC format

Fig 1 shows the HDLC frame format consisting of 6 different fields. Each frame has a different length to serve its respective purpose as following:

- Flag field is an 8-bit sequence that identifies the beginning and the end of a frame. It serves the purpose of synchronization.
- Address field varies from 1 byte to several bytes long and holds the address of the secondary station.
- Control field is a 1 byte or 2-byte field that serves the need for flow and error control.
- Information field contains the actual data to be transmitted by the sender. This data should be correct when received by the receiver.

- Frame Check Sequence (FCS) field can be 2 to 4 byte long and is used for error detection. This field can also be used for error correction.

B. Use of CRC

Cyclic Redundancy Check (CRC) is based on cyclic error-correcting codes. It requires CRC generator polynomial which becomes the divisor in the polynomial division process where the message is treated as the dividend, the quotient is discarded and the remainder is the result. CRC finds a check value which is a fixed-length binary sequence that can be sent with the data, stored or appended to that data. This codeword is generated at the transmitter as well as the receiver. In the case of HDLC, the CRC generator is of the same length as that of the message bits in the information field. Hence the polynomial division process can be used as an addition (without carrying) between the message bits and the CRC generator. The change in the remainder calculated at the receiver shows that the received message bits are corrupted and the comparison of the remainders of transmitter and receiver helps to correct the received bits.

II. Literature Review

HDLC is one of the major protocols in the communication system. FCS field of the HDLC frame has been normally used for the detection of the error in the data [3]. HDLC procedures have been implemented on hardware based on Field Programmable Gate Arrays (FPGAs) [4] [5]. Frame Check Sequence of the HDLC frame has been generated using FPGAs [6]. For high performance, the HDLC controller is developed for data-oriented, switched, and non-switched packet transmission. HDLC protocol has been designed and implemented based on FGPA in Train Communication Network [7].

In [8], the technique of an encoder and decoder is used as error detection and correction code. A fault-tolerant memory system is produced which can tolerate errors, and also support logic operations such as encoding and decoding. But such methods employ coding of data in ‘write’ operations and decoding of data in ‘read’ operations. It leads to the accumulation of data which depends on the frequency of reading and writing application requests. To avoid accumulation, extra logic is used to constantly detect and correct bits in all coded data.

In [9], the Horizontal vertical diagonal (HVD) method is employed to detect and correct RAM errors. HVD help in making the device power efficient in single-error correcting, double error-detecting checker circuits. 2-d parities are used to calculate parity for each row, column, and diagonal. The parity bits are calculated at both transmitter and receiver, and these parity bits are compared to detect errors. The code at the receiver corrects the corrupted data.

In [10], the Horizontal vertical diagonal (HVD) parity check has been used for detecting and correcting errors in HDLC. The four directions in the data part use parity codes to make sure that data is reliable and error bits are detected. It provides a high detection coverage and correction of up to three bits can be done. But if the parity bit is erroneous then the process to detect error becomes complicated. This method can be used in combination with other detection and correction methods to provide high correction coverage.

In [11], the HVD code is used in the HDLC protocol to detect and correct errors. This method employs the use of horizontal (H), vertical (V), and two directions diagonal (D) parity bit to increase detection ability. Based on the calculated parity bits of each dimension, an additional parity bit is computed. For implementing this, the data and, parity bits are considered as a whole world that can be viewed as a 3-dimensional array. To correct the error bits, the first step is to mark the candidate bits that are located where at least two to four erroneous parity lines intersect. The candidate bits are checked to find corrupted bits among candidate bits. The candidate bit where four lines intersect is termed as error bit otherwise the bit is marked correct. To correct the data, all the error bits are flipped.

In [12], the hamming code is used to detect the error in the whole word which consists of data bits as well as parity bits across the length of the array. When the error is found, it is detected whether the error is in the data bit or parity bit. Some extra data is sent with the message, which is used to check the consistency of the received message and correct only 1-bit data.

In [13], the hamming method detects a 2-bit error but corrects 1-bit error only. It is helpful in error detection in case of a few random mistakes or errors. It uses the logic operation of Exclusive-OR to detect the error. Depending on the length of the data, multiple pieces of check bits are inserted into the data. The Hamming code must be at least 4 bits and input and output data must be of 2n power with n greater than the Hamming code.

In [14], the hamming code is implemented in the HDLC protocol to improve the performance of data correction. The Hamming parity technique for HDLC can be optimized using a Hamming parity calculator and an 8*8 RAM. In this method, the data to be transmitted is read from a stored RAM location and parity bits are added to it to create an HDLC frame generator. At the receiver end, the frame is passed through a Hamming parity calculator which first checks the parity of the data and stores it in parity storage. This results in error correction up to 8 bits and bit overhead and code rate of 50% and 66.6% respectively.

In [15], CRC error detection methods have been used in Wireless Sensor Networks. Five CRC divisor bits of CRC-4 are used to enhance the error detection capability in WSN where 100000 frames of 32 bits each show a minimum undetected erroneous frame when the divisor bit used is 10011 in CRC-4.

Fig 2 compares some of the error detection and correction models used for various protocols. It contains the author’s name, the technique followed in the research paper, and some important points related to that particular research.

Author	Technique	Advantages	Limitations
VM Rama Priya [8]	Encoder and Decoder	Fault-Tolerant memory system which is capable of tolerating errors	Increase in redundant data
Narinder Pal Singh [9]	HVD Implementati-on for RAM errors	Less Computational Complexity, Large combination of multiple faults can be corrected	Can correct only single error in a particular code line
Mostafa Kishani [10]	HVD against soft errors	Correction up to 3 error bits	Larger computing operations required to correct and detect error bits
Shubham Fadnavis [11]	HVD based error detection and correction in HDLC protocol	Reduced complexity and requires simpler calculations	Less data correction rate
Dr. Anil Kumar Singh [12]	Error detection and correction using Hamming Code	Less complexity	Correction of only 1 bit error
Varinder Singh [14]	Hamming method implementati-on in HDLC protocol	Enhancement of code rate and reduction of bit overhead, error correction up to 8 bits	More complex architecture
Michael O. Ezea [15]	CRC in Wireless Sensor Network	Has lowest undetected erroneous frames	Increased computation-al and architectural complexity

Fig. 2. Comparison of different error detection and correction models

This proposed model focuses on not only detecting the error in the information field but also correcting it by using CRC to compare the remainders and making a high efficiency error correction technique in HDLC.

III. Proposed Sytem

Fig 3 presents the flowchart of the algorithm used to correct errors in the data. Before the actual transmission of the information, some steps are taken at the transmitter side to achieve the required data sequence to be sent as FCS. This frame not only serves the purpose of detecting the errors in the data if any but also correcting them by comparison process.

The data to be transmitted in the HDLC frame is XORed with the CRC generator to create a remainder which is a result of addition (without carry) of the former two. This remainder is sent in the FCS field of the HDLC frame along with the information to be transmitted. The transmission via any media may result in the insertion of bits, flipped bits, and lost bits. Before processing the information received at the receiver side, it is determined whether the data is correct or not. This is done again by XORing the received data by the predefined CRC generator used by the sender. If the remainder of such operation is the same as the remainder sent by the transmitter, it means that the data is correct. But any change in the received data will compute a different remainder.

When this remainder is compared with that present in the FCS frame, it will point to a specific location where the bit change in the remainders has occurred. This not only informs us about the data being incorrect but also locates the bit which was changed during transmission. Knowing the location of the changed bit in the two remainders helps to correct the data by simply flipping that specific bit. Such a method corrects the data without requesting the transmitter for the ARQ mechanism [16] [17].

IV. Simulation Results

Xilinx-9.2i is used for synthesizing the proposed system and the simulation is done in Xilinx ISE Simulator. An internal signal is used to induce an error in the transmitted data. This makes sure the data received at the receiver is not the data transmitted. It results in the remainders of the transmitter and receiver to be different. A comparison of these remainders will point out the bit that has been corrupted which is solved by simply flipping that particular bit.

A CRC can be used not only for detecting but also locating the error bit in the received data in HDLC protocol. Such a method focuses on correcting the data at the receiver without sending any ARQ to the transmitter. So there is no need of sending the data again even when the data received is incorrect. Such incorrect data can be corrected by the receiver using CRC.

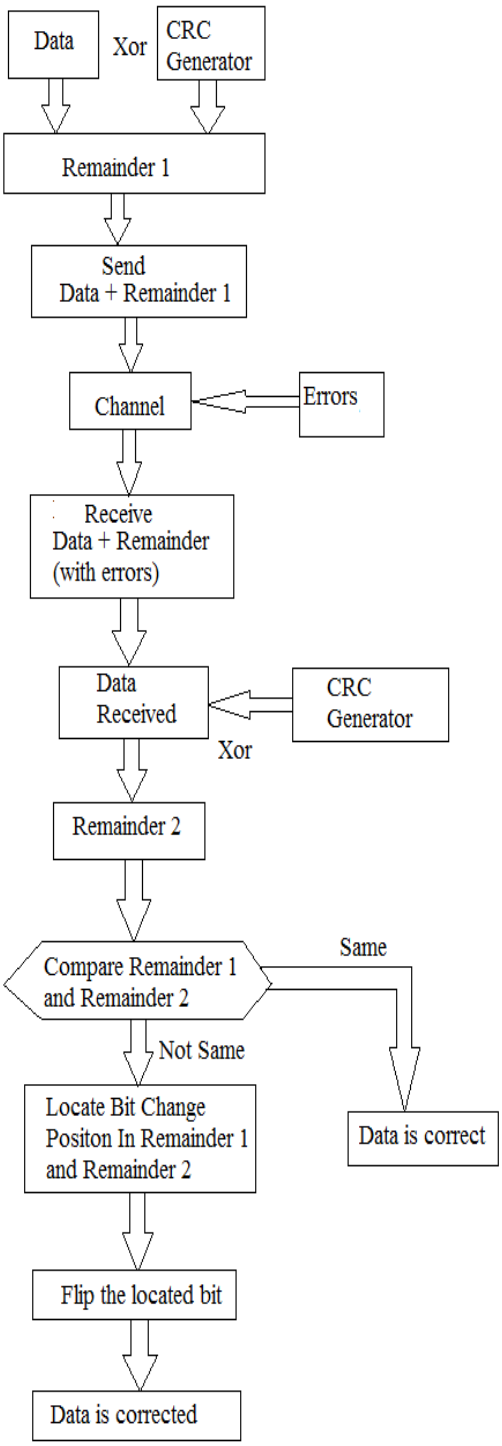

Fig. 3. Algorithm for CRC based HDLC error correction

Fig 4 shows the test bench waveform of the proposed algorithm produced in the Xilinx simulator at 1000ns. It is a graphical environment that includes an input, a stimulus, and the test bench length. It consists of 8-bit input data represented as 8'hCB and 8 bit CRC (8'hA3) which generates a remainder of 8 bits. An internal signal is used to induce an error in the data which is observed in 'Data received'. This

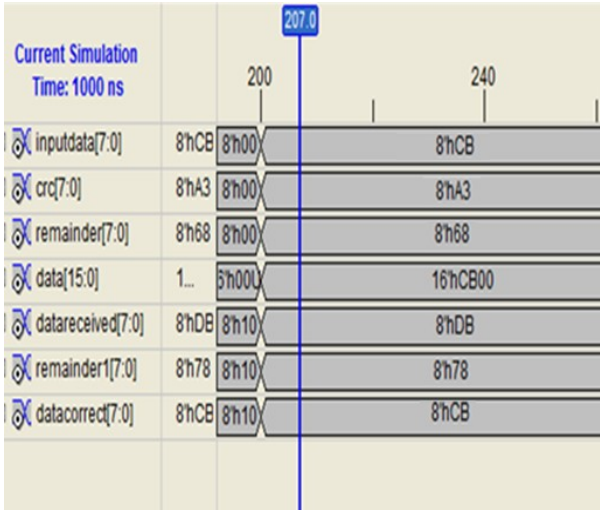

Fig. 4. Test Bench Waveform in Xilinx Simulator

will change the remainder calculated at the receiver. 'Data correct' in fig 4 represents the correction of data even when an error is induced in the original data. Such a mechanism leads to different remainder such that the remainder at the transmitter is 8'h68 while at the receiver is 8'h78.

For the hardware implementation of such a system, it requires a definite number of components. The device utilization summary shows how much area is required by such an algorithm. Fig 5 represents the device utilization summary of the proposed system defining the number of slices, number of 4 input look-up tables or the LUTS and number of bonded input-output blocks (IOBs) required for hardware implementation. Within 5 slices a total of 10 look-up tables are used from 3840 to implement the function generators in Configurable Logic Blocks. 64 IOBs are utilized to implement the input-output functions of such a system.

Fig 6 defines the Register-Transfer Level (RTL) schematic which is the graphical representation for such an algorithm defining the hardware implementation. It shows different blocks, their inputs and, outputs used to implement such a system in a practical environment. It models the flow of digital signals in the hardware and how logical operations are performed on those signals. The input data of 8 bit is passed through a CRC generator to create a CRC remainder which is also 8 bit. The input data is then induced with errors to create corrupted data. This corrupted data is also sent through the CRC generator to generate the CRC remainder at the receiver. Each bit in data has a different block assigned to it that helps to check if the data is correct or not and perform the logical operations on the binary data. Each block has 3 inputs, one for erroneous data, and one for CRC remainder at the transmitter, and one for CRC remained at receiver. For each bit, the CRC at the transmitter and receiver are compared and in case of different remainders, that particular corrupted bit is flipped. If the remainders are the same no changes will be made to the input bit. At the output end, all the bits from the logical blocks are combined together to form the correct data at the receiver end. By this

Device Utilization Summary (estimated values)		
Logic Utilization	**Used**	**Available**
Number of Slices	5	1920
Number of 4 input LUTs	10	3840
Number of bonded IOBs	64	141

Fig. 5. Device Utilization Summary in Xilinx Simulator

method, errors up to 8 bits are detected and corrected as CRC is 8-bit long.

V. Conclusion And Future Scope

A CRC can be used not only for detecting but also locating the error bit in the received data in HDLC protocol. Such a method focuses on correcting the data at the receiver without sending any ARQ to the transmitter. So, there is no need of sending the data again even when the data received is incorrect. Such incorrect data can be corrected by the receiver using CRC. Errors up to the length of CRC can be detected and corrected by this method. For 8 bit CRC, errors up to 8 bit can be corrected. If the length of the CRC is long, the error correction rate will increase linearly. But longer CRC code can also increase the computational and architectural complexity.

Methods can be developed to break the information into blocks so that the CRC generator used can have less length than the information sent. Such methods can be of great use in those systems where the data rate is high. Breaking the information into blocks not only makes the system faster but also reduces the memory requirements for storing the CRC as the length of the generator is reduced. Similarly, CRC codes can be implemented in different communication protocols according to our requirement to correct the data at the receiver end. CRC codes can also be implemented with other correction techniques to achieve a higher performance rate.

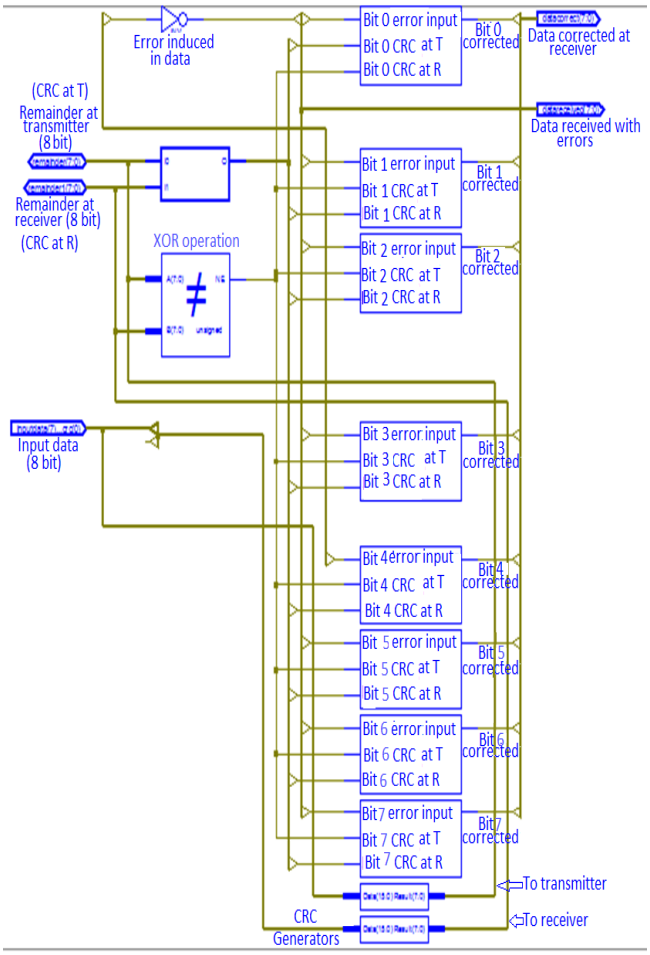

Fig. 6. RTL Schematic in Xilinx Simulator

References

[1] Jatinder Singh and Jaget Singh, *"A comparative study of error detection and coding technqiues"*, International Conference on Advanced Computing and Communication Technologies, Jan 2012

[2] Ku. Rupal P. Bende, A.P. Bagade, S.R, Salwe,"*Review on Design of HDLC Protocol using HDL*", IJIREEICE, Vol 4, Feb 2016

[3] K. Sakthidasan, Mohammed, "*Design of HDLC Controller Using VHDL"*, Internation Journal of scientific and engineering research, Vol 2, March 2011

[4] Sarika G. Joshi, Vaishali S. Dhongde, "*HDLC Protocol Implementation Using VHDL"*, IJIRSET, Vol 3, April 2014

[5] Priyanka Mishra, "*Study and Performance Evaluation of Xilinx HDLC Controller and FCS Calculator*", IOSR Journal of Engineering, Nov 2012

[6] Kshitij S. Patil, G.D. Salunke, Bhavana L. Mahajan, A.S.Hiwale, "*Implementation of HDLC Protocol Using FGPA"*, International Journal of Engineering Science anf Advanced Technology, Vol-2 , 2013

[7] Gaurav Chandil, Priyanka Mishra, *"Design and Implementaion of HDLC Controller Using CRC-16*", IJMER Vol. 3 , 2012.

[8] V.M. Ramaa Priyaa, "*Error Detection and Correction In Encoder and Decoder for Nanmemory*", IJAREEIE, Vol 2, June 2013

[9] Narinder Pal Singh, Sukhjit Singh, "*RAM Error Detection and Correction using HVD implementation*", European Scientific Journal , Nov 2013

[10] Mostafa Kishani, Hamid R. Zarandi, Hossein Pedram, Alireza Tajary, Mohsen Raji, Behnam Ghavami, *"HVD: horizontalvertical-diagonal error detecting and correcting code to protect against with soft errors",* Design Automation for embedded systems, April 2011

[11] Shubham Fadnavis, "*An HVD based error detection and correction code in HDLC Protocol used for communication"*, IJARCCE, Vol 2, Issue 6, 2013

[12] Anil Kumar Singh, "*Error Detection and correction by hamming code*", ICGTSPICC, June 2017

[13] Robbi Rahim, "*Bit Error Detection and Correction with Hamming Code Algorithm*", IJSRSET, Vol. 3, 2017.

[14] Varinder Singh and Narinder Sharma,"*Improving Performance Parameters of Error Detection and Correction in HDLC Protocol by using Hamming Method*", International Journal of Computer Applications, Vol 12, Sept 2015

[15] Michael O. Ezea, "*Performance Analysis of CRC Error detection technqiue in the Wireless Sensor Network"*, International Research Journal of Engineering and Technology, Vol-7, June 2020.

[16] Giuliano Benelli,"*ARQ Protocols for High Efficieny Digital Communication Systems*", IETE Journal of Research, June 2015

[17] Pramod S P, Akshay S kotain, Rajagopal A, "*FPGA implementation of one and two bit error correction using CRC",* International conference on recent trends in computer science and engineering 2012.