# An Efficient Way of Generating CRC bit for Serial data using any polynomial

**Shan Mary Cherian[*], Naveena S George[**], Sherin Mary Enosh[*],Sunitha S Pillai [*]**

[*] Electronics and Communication, St.Joseph's College of Engineering and Technology ,Palai
[**] Electronics and Communication, Mar Athanasious College of Engineering,Kothamangalam

***Abstract-*** CRC is Cyclic Redundancy Check, it is used for generating check bits for detecting data transmission errors in the digital communication networks. The 9-bit CRC polynomial is used for the generation of CRC value . In this paper we have implemented, the generation of CRC value for a 16 bit serial data by using 9 bit polynomial. The CRC bit can be generated for any 16 bit data with any 9 bit polynomial at the time of transmission.

***Index Terms-*** CRC polynomial, Cyclic Redundancy Check,Error detection codes.

## I. INTRODUCTION

There are many techniques in digital communication for generating check bits in order to detect errors occurred in data transmission. Check bits generated will be sent with the message. One of the simplest methods is adding a bit called 'parity bit' to the message[1]. A parity bit is the bit which contains the details that how ,any number of ones are present in the message .i.e, even or odd parity. If a single bit gets changed during the transmission this will change the parity of message from odd to even or vice versa. The sender generates the parity bit by exclusive or'ing (XOR ing) the message bits together. After generating the parity bit ,it then appends the parity bit to the transmitted message. The receiver can check the message by taking the sum of all the bits present in the message including parity and check whether the result is 1, if odd parity is being used. The parity bit technique can be used only for detect 1-bit errors or odd number of errors. Even if this method requires a simple hardware, it cannot be used for complex applications because it cannot detect even bit errors. Another method for generation of check bits is checksum method[2]. There are numerous algorithms for generating checksum bits.

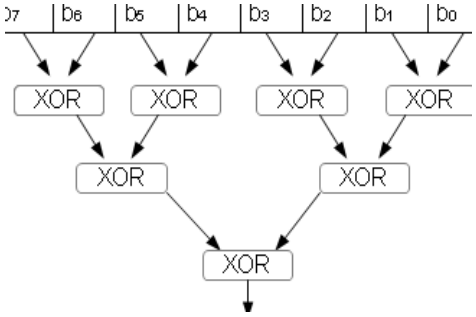


Fig 1: Parity bit generation

Another method for generating check bits are Cyclic Redundancy Codes (CRCs). CRCs can be characterized as one of the most widely used methods for error detection process. But most of the applications are not making use of the original potential offered by CRCs. One of the main reasons for this situation is the absence of effective researches going on in this field. So in order to improve this situation this paper introduces a new method in which any polynomial can be used for generating CRC bits for any serial input data in an efficient way.

## II. BACKGROUND

CRC is basically an international standard which is used for error detection. It makes the data secure by using a checksum or cyclic redundancy check. CRC was first introduced by the CCITT (Comité Consultatif International Telegraphique et Telephonique) which is now known as ITU – T (International Telecommunications Union). The CRC method performs nothing but simple binary division. What happens exactly in CRC is that a sequence of redundant bits called the CRC remainder is appended to the end of the original data stream prior to the transmission. The data thus obtained becomes perfectly divisible by another predetermined CRC polynomial. When it reaches its destination, the received data bits are then divided by the same number and ensure that the remainder is zero.

A CRC calculation can be mathematically described as a polynomial division which is performed over the input data by a generator polynomial called G(x) [3]. G(x) is commonly called as CRC polynomial or feedback polynomial. The remainder obtained after this division is then appended with data. This appended value is known as CRC bit. The error detection calculation is done at the receiver side and if the CRC bit obtained by the sender and receiver are different, then an error is detected. For obtaining an eight bit CRC value a nine bit polynomial is used.

## III. METHODOLOGY

In the previous researches the CRC value is calculated by using a predetermined polynomial which has been known by both the sender and receiver. In this approach the CRC value is calculated using a technique called LFSR (linear feedback shift register) which enables the calculation of the CRC value by any polynomial [4]. This helps the system to work in a more efficient way because here the sender have to set the polynomial according to the serial input data. The LFSR is an efficient way for generating CRC value. LFSR is built by using D flip-flops and Exclusive-OR gates. For serial data input the CRC is

calculated by using the LFSR. The figure 2 given below shows the general representation of LFSR.

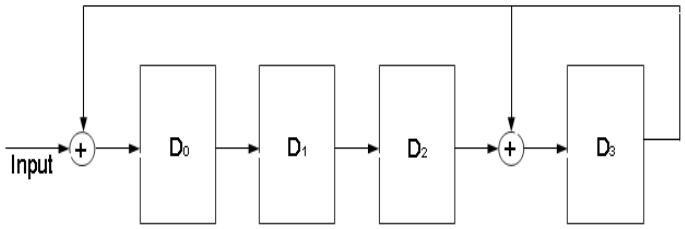


Fig 2:Linear Feedback Shift Register.

In this method the input data is inserted serially into the LFSR and after calculating the CRC value the output is obtained in parallel i.e., it is a serial input parallel output device. First the LFSR will represent the polynomial in the form of binary sequence. For this the coefficients of polynomials are considered, it can be either 0 or 1. The system will perform the XOR function on the basis of coefficients of the polynomials. If the coefficient of the polynomial is zero it will get shifted without performing the XOR operation and if the coefficient of the polynomial is one, before getting shifted it will perform the XOR operation with the shift value from the D flip-flop is performed. After inserting the last input bit the value stored in the D flip-flops are taken as the output CRC value. Here we have used the VHDL language for hardware description.

Now we will describe the CRC calculation of 16 bit serial inputs and we need to generate 8 bit CRC for that. For generating 8 bit CRC value a 9 bit polynomial is needed. The figure 3 showing the generation of CRC value using the polynomial 100110001.

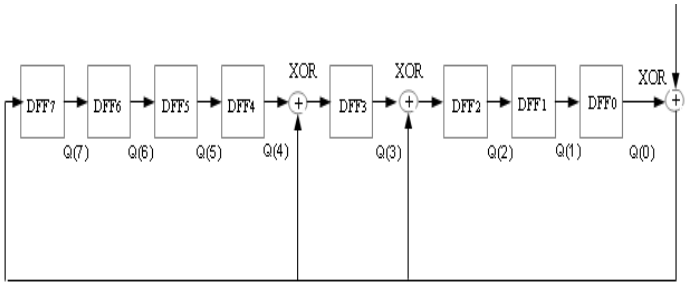


Fig 3:LFSR for generation of CRC for the polynomial 100110001

## IV. RESULTS

The CRC value calculation is implemented using VHDL code and simulated using Modelsim SE 6.5 .The figure 4 shows the simulation result of data FFFF and for the polynomial 100110001.

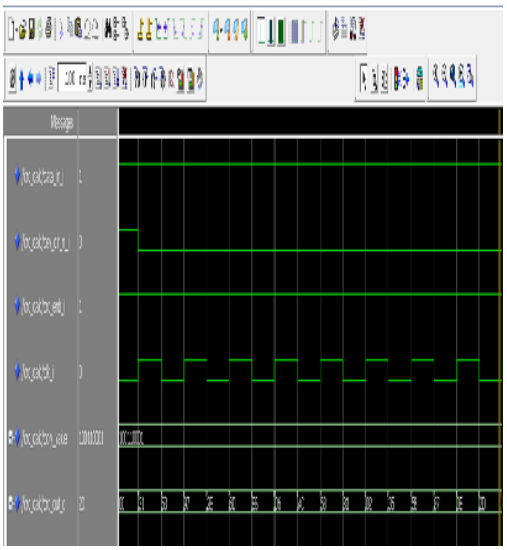

Fig 4: Simulation Result for data FFFF and polynomial 100110001.

The figure 5 shows the simulation result of data FFFF and for the polynomial 111010101.

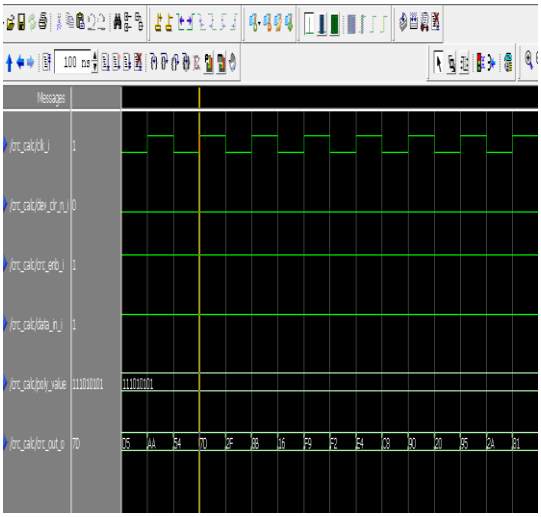

Fig 5:Simulation Result for data FFFF and polynomial 111010101

## V. CONCLUSION

By using this method we can generate CRC bits for any polynomial for any serial data. Therefore more efficient polynomial can be used for generation of CRC bits and hence reduce the possibilities of undetected errors.

## REFERENCES

[1] "Error detection and Correction ," Logic and Computer Design Fundamentals 3rd Edition1.

[2] Theresa Maxino, "Revisiting Fletcher and Adler Checksums ," *DSN* 2006 *student forum* .

[3] Philip Koopman,Tridib Chakravarty , "Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks," The International Conference on Dependable Systems and Networks, DSN-2004

[4] Kewal K Saluja, " Linear Feedback Shift Registers theory and applications" University of Wisconsin-Madison,updated in 1991.

## AUTHORS

**First Author** – Shan Mary Cherian, pursuing M.Tech from St Joseph's College of Engineering and Technology, Palai. shanmary04@gmail.com.

**Second Author** – Naveena S George, pursuing M.Tech from Mar Athanasious College of Engineering, Kothamangalam, nsgeorge02@gmail.com .

**Third Author** – Sherin Mary Enosh, pursuing M.Tech from St Joseph's College of Engineering and Technology, Palai.

**Fourth Author –** Sunitha S Pillai ,Assistant Proffesor ,ECE department St Joseph's College of Engineering and Technology, Palai.

**Correspondence Author** – Shan Mary Cherian, shanmary04@gmail.com,shanmarycherian@gmail.com,+919447 778142.