FPGA IMPLEMENTATION OF 4-BIT PARALLEL CYCLIC REDUNDANCY CODE

Hemant Sharma[1], Samyak Tomar[2], Jitendra Kanungo[3]

[1,2,3]*Department of Electronics and Communication Engineering, Jaypee University of Engineering and Technology, Guna, India*

[1]***hemant10192@gmail.com***

Abstract

This paper presents a different approach to solve the parallel CRC circuit. The previous works have been studied. Certain drawbacks were observed in the previous works. Some authors had used Linear feedback shift registers to do serial implementation. This approach resulted in a circuit that was inefficient in terms of time utilization. Though some authors had used the concept of parallel CRC but there was scope for improvement. We have worked on the related issues and have proposed an efficient mechanism. We have developed the VHDL code using VHDL structural modelling. The work was also compared with existing models of parallel implementation of four bit CRC circuit. The code is written for four bit parallel CRC and FPGA implementation of the code was done. Comparing with existing work, the proposed model is more efficient in terms of hardware utilization. As the hardware utilization has been been done in an efficient way, the overall efficiency of the parallel CRC is found to improve.

Keywords: *CRC,PARALLEL CRC, STRUCTURAL MODELLING*

----------***----------

1. INTRODUCTION

Cyclic Redundancy Check(CRC) is the subset of linear block codes.It is mainly used in communications,data storage,industrial measurement and control system.The evolving world of telecommunications requires increasing reliability and speed in communications [1].

The Cyclic Redundancy Check was invented by W.Wesley. CRC is an error detecting code which is used to detect the error in the message bits. This error is transmitted from the sender to the receiver.The hardware mapping problem of the parallel CRC compuation was first addressed by Braun which was based on the matrix computation technique which is different from Pei.He incorporated both pre- and post- CRC computation circuits so as to achieve 32-bit checksum word at 450 Mbps by using FPGA technology in 1996.

There are two types of codes:block codes and the other is convolution codes.Both of them introduce the redundancy by adding parity bits to the message bits.Cyclic Redundancy Check(CRC) are the subset of cyclic codes which is also the subset of linear block codes.The main applications of CRC is detecting errors in data communication systems and devices for storage.Integrity of blocks of data called Frames is also verified.Here an extra n bit sequence is appended by the transmitter to every frame which is called as Frame Check Sequence(FCS).

2. CALCULATION OF CRC

CRC uses binary division technique.A sequence of redundant bits also known as CRC or the CRC remainder,are appended at the end of the message bits so that the resultant data which include the message bits and the redundant bits is divisible by a second predetermined number which is binary.

Firstly, a string of m 0's is appended at the end of the message bits.The bits that we are adding is m.It is less than the bits in the divisor which is m+1 by one.

Secondly, the resultant bits are divided by the divisor using the binary division method.The remainder generated is the CRC.

Thirdly, the CRC generated is added to the data unit by replacing the redundant bits.

The data unit is checked at the receiver side.Now the data unit which include the message bit and the CRC is again divided by the same divisor known both to sender and receiver by using binary division. If the remainder is zero then message sent is without error else error is there in the message.

3. PROPOSED DESIGN

In 2003,Giuseppe Campebello et al. [6] presented a theoretical result to help in realizing hardware which was high speed.It was for parallel CRC checksums.The authors studied the serial as well as parallel implementation of CRC and found out technical gaps in the research papers. Recursive formula for parallel implementation of CRC was derived by the authors. Then , a VHDL code was generated by the authors for the logic equation for CRC .The VHDL code generated was done for the first time by any researcher .In comparison with previous works, this new scheme was faster and more compact and was not dependent on the technology used in its realization. However, the VHDL code which they developed was done by using behavioral modelling. The hardware utilization was not efficient in this case. We developed the VHDL code for the parallel circuit

using structural modelling for 4 bit CRC.Our code is efficient in terms of hardware utilization in comparison with the existing work.

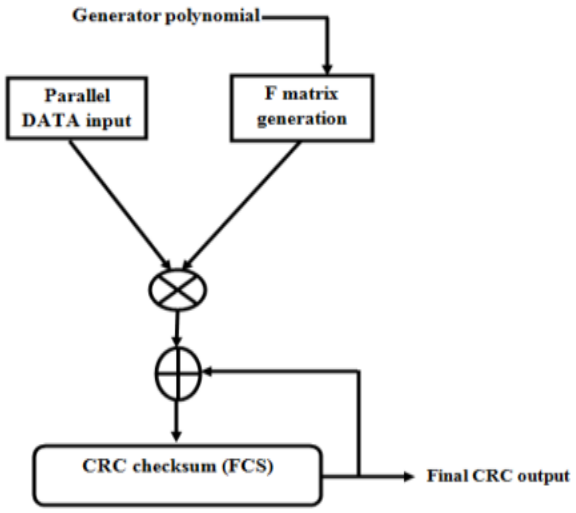
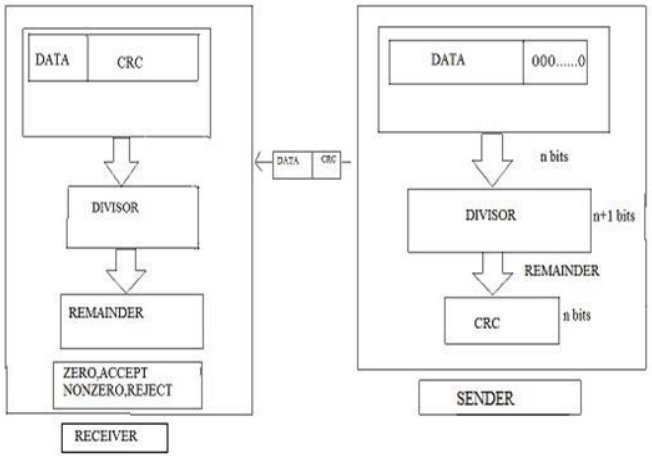


Figure 1

4. PARALLEL CRC COMPUTATION

ALGORITHM

- STEP no 1:The F matrix is generated with the help of generator polynomial.
- STEP no 2:The message is fed as input and the bits are fed parallely
- STEP no 3:These parallel bits are multiplied bitwise with the F matrix.
- STEP no 4:Initially the value of FCS is set as zero by clearing all the LFSRS .
- STEP no 5:The output obtained from STEP3 is Exored with the previous value of FCS .
- STEP no 6:Then new FCS is obtained which is fed back.
- STEP no 7:Now the new value of FCS is calculated by Exoring the output from STEP3 and the FCS value which is fed back.
- STEP no 8:The final value of FCS will be obtained after (k+m)/w cycles.



5. COMPARISON WITH EXISTING WORK

The VHDL code developed by Giuseppe Campebello et al. [6] and the code developed by us was run on the same platform. The below mentioned result was obtained.

As we can see that clearly the hardware utilization is better in our case.Thus the hardware complexity of the circuit gets reduced on implementation of our code.

HARWARE COMPONENT UTILIZATION	OUR PROPSED DESIGN	REPORTED DESIGN
REGISTERS	1%	1%
LUTs	10%	70%
IOBs	60%	_
BUFFER	_	
LUT-FF	_	69%

6. RTL SCHEMATIC

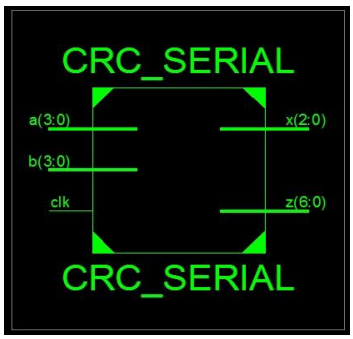


Fig 2. RTL Schematic of serial CRC

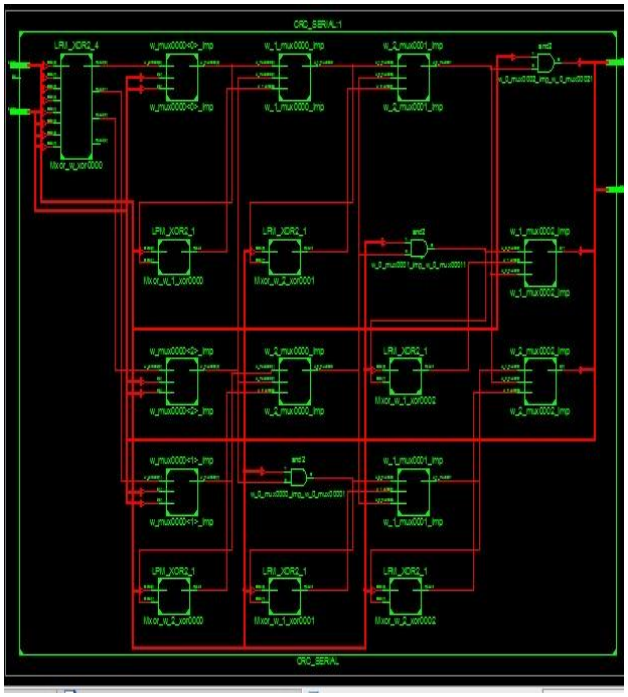


Fig 3. RTL Schematic of parallel CRC

7. CONCLUSION AND FUTURE SCOPE

Calculation of CRC forms an important aspect in determining the speed of any communication system. We studied the various methods for generation of CRC. We also analyzed the work done by various researchers in this field .We found out that a lot of work has been done in this field regarding speed of the circuit, choosing generator polynomial,hardware architecture etc. Some of the work of early researchers was focused on serial generation of CRC.However, later it was realized that this was not an efficient method. So,later the work was shifted towards the parallel architecture. In our work, we have focused on developing an efficient VHDL code for the parallel implementation of the CRC circuit. We developed VHDL code using serial as well as parallel architecture and found out the time delay in both the cases. The delay was reduced to almost half in the parallel architecture and thus we developed an efficient code for optimizing the speed of the circuit.

Out of the various research papers, we found that some researchers were able to optimize speed by minimizing hardware, while some achieved the desired performance by improving their code.Initially Serial Linear Feedback Shift Registers were used to design the CRC code. Later on, it was realized that using Serial LFSR had its own limitations. It could process only one bit at a time in a single clock pulse. So optimum speed was not achieved. To overcome this limitation parallel implementation of LFSRs was done by many researchers. As a result the speed of the circuit improved greatly. It was also realized that while the parallel implementation of LSRs increased the spix eed but the method could not be customized for different CRCs. Although some researchers achieved good speed as high as 35-40 Gbps. However, the method was applicable only for particular CRC. On the other hand, some researchers did develop codes for CRCs [9] of different length but the desired speed could not be achieved.

The F matrix which we have used in our code is only applicable for four bit CRC.
However ,the code for F matrix for higher bit CRCs, rather a generic code for matrix can be written. Also the hardware complexity can be further minimized using loops to minimize the code.

REFERENCES

[1] Gaurav Chawla and Vishal Chaudhary, "FPGA Implementation of Cyclic Code Encoder and Decoder," Advance in Electronic and Electric Engineering.

[2] Rameshwr T. Murade ,MD. Manan Mujahid,M.A.M. Sabir, "The Design and Implementation of a Programmable Cyclic Redundancy Check Computation Circuit Architecture Using FPGA,"*International Journal of Science and Modern Engineering(IJISME)*,Nov. 2013.

[3] S.V.Viraktamath ,Ms. Veena Joshi ,MS Usha Nagesh Naik, Dr. Girirsh V. Attimarad, "Implementation of CRC on FPGA,"*International Journal of Scientific Research*, Nov. 2013.

[4] B. Ramya Sree , B. Manjula, K. Murali Krishna, B. V. Rama Mohana Rao, "Analysis of an error detecting code in block based transmission,"*International journal of Communication Engineering Applications-IJCEA*,July-Dec. 2011.

[5] Wael M El- Medany et al., "FPGA Implementation of CRC with Error Correction," ICWMC 2012 : *The Eighth International Conference on Wireless and Mobile Communications.*

[6] Giuseppe Campobello, Giuseppe Patane , and Marco Russo, "Parallel CRC Realization,"*IEEE Trans. of Computers*,Vol. 52, No. 10,Oct. 2003

[7] P. Sushma, N.Sushma, V.Sowmya, B Krishna,D. Arun Kumar, "64-bit parallel CRC Generation for High Speed Applications," in *International Journal of Computer Applications & Information Technology*,Feb.-March 2014

[8] Lin and Castello, Error control coding , Pearson Education; Inc. 2004.

[9] Volnei A. Pedroni, Circuit Design and Simulation with VHDL, MIT Press; 2nd Revised edition edition.

[10] R.P. Jain, Modern Digital Electronics, McGraw Hill Education (India) Pvt Ltd, 3rd edition.