IJESMR

International Journal OF Engineering Sciences & Management Research

A REVIEW OF DESIGN AND SIMULATION OF PARALLEL CRC GENERATION ARCHITECTURE FOR HIGH SPEED APPLICATION

Deepali P. Atrawalkar[1]
[1]Digital Electronics, G.H.R.I.E.M., Jalgaon, India
Correspondence Auther: deepaliatrawalkar@gmail.com

KEYWORDS:CRC Generation, CRC Architecture FPGA, VHDL.

ABSTRACT

High speed data transmission is the current scenario in networking environment. Cyclic redundancy check (CRC) is essential method for detecting error when the data is transmitted. With challenging the speed of transmitting data, to synchronize with speed, it's necessary to increase speed of CRC generation. Starting from the serial architecture identified a recursive formula from which parallel design is derived. For simulation and functional verification we will use modelSim and alteraQuartus 2. High speed data transmission is the current scenario in networking environment. Cyclic redundancy check (CRC) is essential method for detecting error when the data is transmitted. With challenging the speed of transmitting data, to synchronize with speed, it's necessary to increase speed of CRC generation. Starting from the serial architecture identified a recursive formula from which parallel design is derived. This paper presents 64 bits parallel CRC architecture based on F matrix with order of generator polynomial is 32

INTRODUCTION

Cyclic redundancy check is a popular error detecting code computed through binary polynomial division. CRC is used in data communication and various fields such as data storage, data compression dealing with data errors .Many times speed requirement makes software schemes impractical and demand a dedicated hardware. CRC is a very powerful and easily implemented technique to obtain data reliability.
CRC can generate in 2 ways:
1. Serial CRC generation
2. Parallel CRC generation

Usually the hardware implementation of CRC computation is based on the linear feedback shift register (LFSR), which process data in serial way. But we cannot achieve a high throughput by serial calculation of CRC code. Instead of it parallel CRC generation overcomes this problem. The parallel method process whole data words on cascading the LFSR. Cyclic redundancy check is commonly used in data communication and other fields such as data storage, data compression, as a vital method for dealing with data errors [6]. Usually, the hardware implementation of CRC computations is based on the linear feedback shift registers (LFSRs), which handle the data in a serial way. Though, the serial calculation of the CRC codes cannot achieve a high throughput. In contrast, parallel CRC calculation can significantly increase the throughput of CRC computations. For example, the throughput of the 32-bit parallel calculation of CRC-32 can achieve several gigabits per second [1.]

LITERATURE REVIEW

For the transmission of high speed data, parallel CRC generation technique is presented by Hitesh H. Mathukiya [1] and Naresh M. Patel [1].This paper presents 64 bits parallel CRC architecturebased on F matrix with order of generator polynomial is 32.Proposed design is hardware efficient and required 50% lesscycles to generate CRC with same order of generator polynomial. The whole design is functionally verified using Xilinx ISE Simulator.

VLSI Implementation of Parallel CRC Using Pipelining, Unfolding and Retiming is presented by Sangeeta Singh, S. Sujana, I. Babu, K. Latha [3].

A new hardware scheme for computing the transition and control matrix of a parallel cyclic redundancy checksum is proposed by Martin Grymel and Steve B. Furber [5] in the IEEE paper. This opens possibilities for parallel high-speed cyclic redundancy checksum circuits that reconfigure very rapidly to new polynomials. The area requirements

IJESMR

International Journal OF Engineering Sciences & Management Research

are lower than those for a realization storing a pre computed matrix. An additional simplification arises as only the polynomial needs to be supplied. The derived equations allow the width of the data to be processed in parallel to be selected independently of the degree of the polynomial. The new design has been simulated and outperforms a recently proposed architecture significantly in speed, area, and energy efficiency**.**

This paper presents implementation of parallel Cyclic Redundancy Check (CRC) based upon DSP algorithms of pipelining, retiming and unfolding [8]. The architectures are first pipelined to reduce the iteration bound by using novel look-ahead techniques and then unfolded and retimed to design high speed parallel circuits. This paper presents the comparison between the parallel implementation of CRC-9 and its serial implementation. It also shows that parallel implementation uses less number of clock cycles than the serial implementation of CRC-9 thereby increasing the speed of the architecture.

PROPOSED WORK

The CRC is a short fixed-length checksum for an arbitrary data block. It will accompany the data and can be validate at an endpoint through recalculation . Differences between the two CRC values indicate a corruption in either the data or the received CRC itself. The hardware implementation of CRC5 is implemented as shown in the diagram:

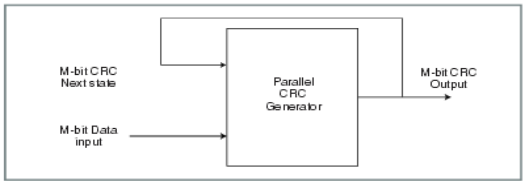

Fig.1: Hardware CRC5 implementation

3.1 SERIAL CRC

The generation of serial CRC is based on the linear feedback shift register (LFSR). The operation of LFSR for CRC calculation is nothing but the binary divisions. Binary divisions generally can be performed by a sequence of shifts and subtractions. Inmodulo2 arithmetic the addition and subtraction are equivalent to bitwise XORs and multiplication is equivalent to AND. Fig. 1 illustrates the basic architecture of LFSRs for serial CRC calculation.

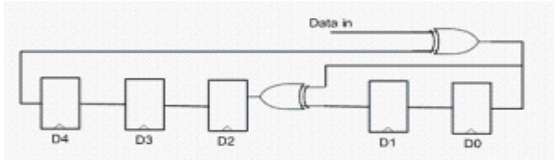

Fig. 2: Serial CRC generation using LFSR

3.2 PARALLEL CRC

An another method of generating CRC code is parallel CRC generation. In this technique the next state CRC output is a function of the current state CRC and the data.

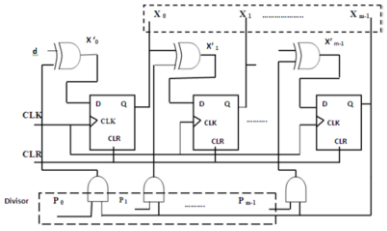

Fig.3: Parallel CRC block

IJESMR

International Journal OF Engineering Sciences & Management Research

There are different techniques for parallel CRC generation as:

1. A Table –Based Algorithm for Pipelined CRC calculation
2. Fast CRC Update
3. F matrix based parallel CRC generation
4. Unfolding, Retiming and pipelined Algorithm

In computing, a pipeline is a set of data processing elements connected in series, so that the output of one element is the input of the next one. Our fast CRC update method is extended from the parallel CRC calculation and can adapt to a number of bits processed in parallel. The method can also reduce the data traffic and power consumption of the CRC calculation unit.Retimingis the technique of moving the structural location of latches or registers in a digital circuit. Unfolding is a transformation technique of duplicating the functional blocks to increase the throughput.

CONCLUSION

32bit parallel architecture required 17 ((k + m)/w) clock cycles for 64 byte data [1][5]. Proposed design (64bit) required only 9 cycles to generate CRC with same order of generator polynomial.

ACKNOWLEDGMENTS

My special thanks to all that experts who have contributed towards development of this research paper.

REFERENCES

1] *Hitesh H. Mathukiya; Naresh M. Patel, "A Novel Approach for Parallel CRC generation for high speed application",2012 International Conference on Communication Systems and Network Technologies.*

2] *Campobello, G.; Patane, G.; Russo, M.; "Parallel CRC realization," Computers, IEEE Transactions on, vol.52, no.10, pp. 1312- 1319, Oct.2003*

3] *Sangeeta Singh, S. Sujana, I. Babu, K. Latha. "VLSI Implementation of Parallel CRC Using Pipelining, Unfolding and Retiming",IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 2, Issue 5 (May. – Jun. 2013)*

4] *Albertengo, G.; Sisto, R.; , "Parallel CRC generation," Micro, IEEE, vol.10, no.5, pp.63-71,Oct1990*

5] *Martin Grymel and Steve B. Furber, " A Novel Programmable Parallel CRC Circuit", IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS,2010 IEEE*

6] *M.D.Shieh et al., "A Systematic Approach for Parallel CRC Computations," Journal of Information Science and Engineering, May 2001.*

7] *Braun, F.; Waldvogel, M.; , "Fast incremental CRC updates for IP over ATM networks," High Performance Switching and Routing, 2001 IEEE Workshop on , vol., no., pp.48-52, 2001*

8] *Weidong Lu and Stephan Wong, "A Fast CRC Update Implementation", IEEE Workshop on High Performance Switching and Routing ,pp. 113-120, Oct. 2003.*

9] *S.R. Ruckmani, P. Anbalagan, " High Speed cyclic Redundancy Check for USB" Reasearch Scholar, Department of Electrical Engineering, Coimbatore Institute of Technology, Coimbatore- 641014, DSP Journal, Volume 6, Issue 1, September, 2006.*

10] *Yan Sun; Min Sik Kim; , "A Pipelined CRC Calculation Using Lookup Tables," Consumer Communications*

11] *and Networking Conference (CCNC),2010 7th IEEE, vol., no., pp.1-2, 9-12 Jan. 2010*

12] *Sprachmann, M.; , "Automatic generation of parallel CRC circuits," Design & Test of Computers, IEEE, vol.18, no.3, pp.108-114, May 2001 58358*

IJESMR

International Journal OF Engineering Sciences & Management Research

AUTHOR BIBLOGRAPHY

Ku. Deepali P. Atrawalkar
Received the degree in Electronics and Telecommunication from Sant Gadage Baba, Amaravti University.
Now persuing post graduation in Digital Electronics from Jalgaon.
Email: sankmoharir@gmail.com