



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

A Fast and Area Efficient Cyclic Redundancy Check-Based Decoder for Software Defined Radio

R.D.Fating, S.S.Jain

M. Tech, Dept. of Electronics and Communication Engineering, DMIETR, Wardha, Maharashtra, India

Assistant Professor, Dept. of Electronics and Communication Engineering, DMIETR, Wardha, Maharashtra, India

ABSTRACT: The requirement for adaptable and high-speed error decoders has recently increased in software-defined radio-based communication systems. Software Defined Radio (SDR) refers to the technology through which the software module running on general hardware platform like FPGA, DSP, GPP and general purpose microprocessors are used to implement radio functions such as generation of transmitted signal at transmitter and tuning /detection of received radio signal at receiver. Cyclic Redundancy Check based error decoder (CRC) is one of the promising error decoders, shows acceptable accuracy; however, the serial error decoder limits its use in real time communication due to the requirement of lot of time for performance. To address this issue, this paper proposes a fully parallel implementation of the CRC128(Syndrome bits) code on a Field Programmable Gate Array (FPGA) using comparator based architecture by exploiting massive data-parallelism which detect the error in only one clock cycle and provides time efficient communication. In this design Verilog HDL use as a description language for software defined radio (SDR) and simulated using Xilinx ISE Design suite 13.2. This proposed method can provide maximum error detection and correction capability with reduction in delay and the throughput of the proposed architecture reaches up to 316.83 Gbps with about 38% area utilization.

KEYWORDS: Software Defined Radio, Cyclic Redundancy Check, Error Coding, Serial Processing, Parallel Processing.

I. INTRODUCTION

A. Software Defined Radio

Error correction is an inevitable component of wireless communication systems, and existing error decoders are based on dedicated hardware device for particular communication standards, including worldwide interoperability for microwave access (WiMAX, IEEE 802.16), Wi-Fi (IEEE802.11), fourth-generation long-term evolution (4G LTE), wideband code division multiple access (W-CDMA), and global system for mobile communications (GSM) [13]. However, hardware-based decoders offer limited, if any, of the programmability and flexibility needed for versatile decoding algorithms that can flexibly support multiple communication standards to facilitate users. Once a hardware decoder is manufactured for a specific standard, it cannot realize other standards because the hardware decoder has its own coding scheme, data rate, frequency range, and types of modulation [15]. In addition, the manufacturing cost and time-to-market of hardware devices are very high, adding additional hurdles to keep up with rapidly changing technology [14].

To address these issues, software-defined radio (SDR) is an emerging technology offering software alternatives to existing hardware solutions, and it has recently drawn a great deal of attention in the communication research community. SDR is the software implementation of multi-standard and multi-protocol communication systems in a single hardware platform. In SDR, some or all of the physical layer functions are written in software that operates on general purpose processors (GPPs), Field Programmable Gate Array and digital signal processors (DSPs) because these can provide the necessary programmability and flexibility for various SDR applications [8].

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

1. TYPICAL MODEL OF SDR RECEIVER

As shown in Fig.1, the basic block diagram of an SDR receiver consists of three stages [12]. The signal reception is done by the antenna. The first section is the Radio Frequency (RF) front end where the signal is received and down converted to Intermediate Frequency (IF) by mixing the incoming frequency with local oscillator frequency. The IF section samples the signal using an Analog to Digital Converter (ADC).

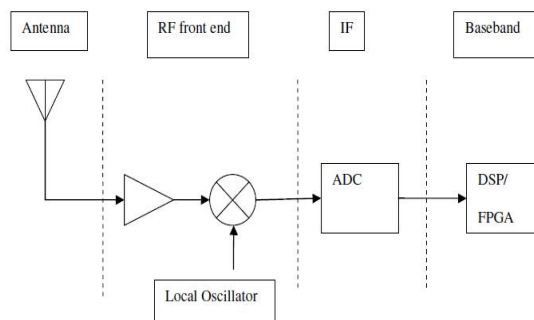


Fig.1. Basic block diagram of SDR

Baseband section is important part of a software defined radio. In this section consist either DSP or FPGA, demodulates and filters the signal. The software or the code is downloaded into the DSP or FPGA hardware, and it produces the output in accordance with the parameters mentioned in the software. For low frequency radios, everything can be done in FPGA, excluding ADC and antenna/power amplifier [12]. So, in this thesis proposed parallel CRC128 architecture that satisfies high speed and throughput for baseband processing in software defined radio communication system.

Error detection and correction methods are used to detect and correct the errors in the data that is transmitted from one point to another. Errors may be introduced in the transmitted data due to noise and external radiations. These errors could be corrected by using various error detection and correction methods that are used for the reliable transport and the reception of data over a communication channel. The errors introduced by the channel should be detected at the receiver end. So the error detection is essential for its correction. If the errors are undetected it may lead to the improper functioning of the system [1]. So several approaches are used for the error detection and correction. This paper deals with the novel method which uses a redundancy based error detection method namely Cyclic Redundancy Check which proposes the parallel comparator based architecture for CRC128 algorithm for finding the error location in incoming 128 bit data in one clock cycle. The throughput of the proposed architecture reaches up to 316.83Gbps with about 38% area utilization.

B. Cyclic Redundancy Check

Error detection means to check whether the received data is correct or not without having a copy of the original message. Error detection uses the concept of redundancy, which means adding extra bits to the end of the data stream for detecting error at the destination. Cyclic redundancy check is the type of redundancy which is calculated from the data which is to be send. CRCs are used to detect the corruption of digital content during production, transmission, processing or storage [2].

- A CRC is a type of a redundancy and it enabled machine calculates a fixed-length binary series, known as the *CRC*, for each block of data to be sent and appends it to the data, forming a *codeword*.
- When codeword is received, the machine either compares its check value with one newly calculated from the received data block, or equivalently, performs a CRC on the whole codeword and compares the resulting check value with an expected residue constant.
- If the CRC check values do not match, then the block contains error.
- The device may take corrective action, such as rereading the block or requesting that it be sent again. Otherwise data is assumed to be error free.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

II. RELATED WORK

There have been successful architectures to accelerate CRC algorithms. The researches can be divided into two categories: serial CRC and parallel CRC. Paper [16] is one of the most popular execution of serial CRC. In its performance, a simple circuit based on shift registers performs the CRC computation by handling the message one bit at a time. Processing one bit per cycle greatly limits the throughput of CRC circuits. So there are many researches generate CRC in parallel. These techniques include:

- 1) Table-Based algorithm
- 2) Fast CRC update
- 3) F-matrix based parallel CRC generation
- 4) Unfolding, retiming and pipelining algorithm

Neepa P. Mathew and Anith Mohan [1] proposes Serial and parallel CRC64 implementation for 128bit of input data. The serial implementation is done by using LFSR based method and the parallel implementation is done by using slicing by 4 algorithm. Here 128 bit of data is divided into four slices where each slice consist of four bytes of data. Then it performs the CRC computation using lookup tables among the divided bytes in parallel. At last, the results are XORed to obtain the CRC64 value of given message for error detection purpose. Hybrid matrix code method is proposed for error correction.

W. Lu and S. Wong [9] proposes a fast CRC update technique. In this technique, It is not necessary to calculate CRC each time for all the data bits, instead one only needs to calculate CRC for those bits that are changed. They calculate the intermediate result of changed fields using the parallel CRC calculation and perform a single step update afterwards. Consequently, the number of cycles needed to recalculate the CRC codes is greatly reduced. They estimate that the theoretical throughput can reach about 56 Gbps assuming the frame size distributions are realistic.

H. H. Mathukiya and M. P. Naresh [3] proposes F-matrix based parallel CRC generation. In this method, data input and each element of the F-matrix generated from given generator polynomial are ANDed, then the result will be XORed with present state of CRC checksum. It presents 64 bit parallel CRC architecture based on F-matrix with the order of generator polynomial is 32. When processing 64 bytes data, their method costs 9 clock cycles.

S. Sangeeta, et al. [11] this paper proposes Retiming is used to increase clock rate of circuit by reducing the computation time of critical path. For example, paper [3] applies unfolding technique to pipelined architecture to increase the throughput of the circuit. Then it applies retiming to the architecture to reduce the critical path delay. Paper [3] also points out that the design is not efficiently applicable for the LFSR architecture of any generator polynomial. It is efficient for the generator polynomials with many zero coefficients between the second and third highest order nonzero coefficients. When processing the same input data, the serial implementation of CRC-9 costs 9 clock cycles while their method costs 5 clock cycles without increasing the hardware cost.

There are advantages and disadvantages for each technique to generate CRC value. Fast CRC update technique requires extra memory to store the old CRC value and data. Unfolding architecture increases the number of iteration bound and the parallel degree is limited. The F-matrix based architecture can improve the parallel degree greatly. But when the input data is more than 64 bit long, the delay of the XOR tree can be the critical path of the circuit. Table based architecture for CRC64 implementation from 128 bit input required a lot of memory and to change 1 LUT all the values of LUT change.

This techniques only used for CRC generation at the receiver side. But receiver requires extra N clock cycle to check every bit and perform error location and correction. For this purpose, this paper proposed the parallel comparator based architecture for CRC128(Syndtome bits) algorithm which can provide the faster error detection and correction with minimum delay, also it provide more throughput than other traditional methods and no work has been reported in literature on parallel comparator based architecture for CRC128 algorithm.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

III. ERROR DETECTION PROCESS

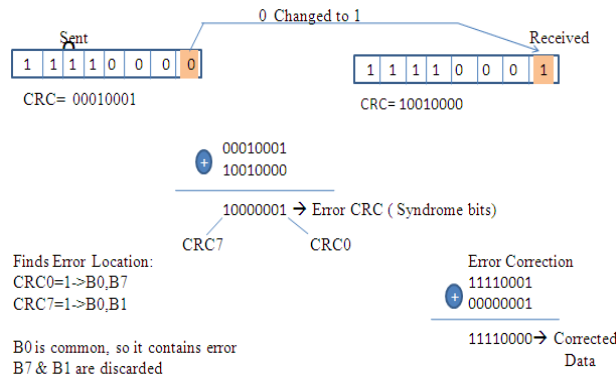


Fig.2. Error detection process

This is the error detection process in which 8bit data with 8bit CRC (0001000111110000) is send to receiver from sender. During the transmission of data, bit 0 is changed to bit 1 and at the receiver side we get the corrupted data. Then again CRC is calculated from the received corrupted data. Then both CRCs i.e.initial CRC and freshly calculated CRC are compared.If they do not match that means error is present in the received data.Then performe XORed operation between both CRCs then we get error CRC(10000001). Then it is necessary to finds the error location. CRC0 and CRC7 are the error CRC, but CRC0 is made by B0&B7 and CRC7 is made by B0&B1.Here B0 is common, so it contains error. B7 & B1 is discarded.

To performing error correction, here taken received corrupted data and error bit i.e. B0 and performing XORing between them. Hence got corrected data.

In this paper, work has been done on error CRC128. For this purpose, parallel processing has been applied on error CRC128 using comparator based parallel architecture.

IV. PROPOSED ARCHITECTURE

1.Serial CRC architecture

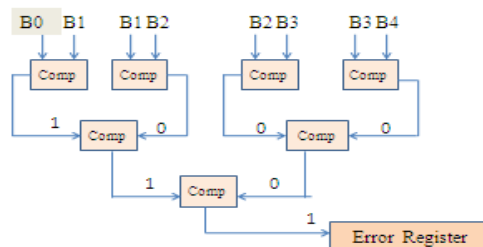


Fig.3. Serial CRC architecture

This is a serial CRC architecture which work for only single bit in 1 clock cycle. Hence it requires N clock cycle for N bit data to check every bit and find error location. Hence it requires 128 clock cycle for 128bit data. As shown in fig 2, CRC0 and CRC7 are error CRCs which are made by B0,B1 or B0,B7 respectively. This bits are taken in this comparator based architecture in serial manner. In one clock cycle, it work only for single bit i.e.B0 and error register decides whether it contains error or not in the form of 1 or 0. If error register contains 1 that means that particular bit contains error. As shown in fig, this architecture firstly work for B0 and error register contains 1 means B0 contains error. After that it work for B1,B2.....B127 in serial manner. But these Serial error decoder limits its use in

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

real time communication. To address this issue, parallel error decoder has been designed, which provide time efficient communication.

2. Parallel CRC architecture

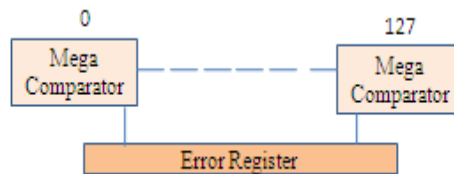


Fig.4. Parallel CRC architecture

This is a comparator based parallel CRC architecture. It works simultaneously for 128 bit data in one clock cycle. For that purpose it accelerate syndrome bits(CRC128). By accelerating it , we can find out error in 128 data bits . Hence by using syndrome bits, we can find out error location in 128 data bits because 1CRC is made by two data bits. So in this manner it works for all 128 data bits in 1 clock cycle. It requires only 1 clock cycle for N bit data to check every bit and finds the error location. For 128 bit data it requires only 1 clock cycle for performance. When 128 bit digital data given to the mega comparator, it compares between 0 and 1 and passed 1 to error register. Then error register detect error bit location. Hence this comparator based parallel architecture for CRC128 algorithm performs only in 1 clock cycle with reduction in delay.

V. PSEUDO CODE

- Step 1: Send beacon signal to transmitter from receiver.
- Step 2: After getting the beacon transmitter generate 128 bit CRC value from 128 bit data.
- Step 3: Both 128bit data and 128CRC appends together and form the codeward of 256 bit.
- Step 4: Codeword sends to receiver from transmitter.
- Step 5: Receiver receive the information and again calculate the 128bit CRC from 128 bit received data .
- Step 6: Both initial CRC and freshly calculated CRC compares and if they do not match means data is errored otherwise error free.
- Step 7: Syndrome bits(Error CRC) are generated by performing the binary operation between the initial redundant bits and new redundant bits.
- Step 8: Accelerate syndrome bits in serial and parallel manner using comparator based architecture and find error bits.
- Step 9: Compare result with reference work(1).
- Step 10: End.

VI. SIMULATION RESULT

In this proposed work, parallel CRC128 architecture has been designed. Which achieves less area (No. of slices), low delay and high operating frequency as compare to reference work[1]. The proposed work was synthesized on on Xilinx ISE Design Suite Spartan3E XC3S100E & Comparison on the basis such parameter is summarized in the table 1.

Table.1 : Comparison of Delay and operating frequency of proposed work with reference work

Sr.No	Parameter	Proposed Work	Reference Work
1.	Delay	4.04ns	6.93ns
2.	Operating Frequency	247.524Mhz	144.300Mhz

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

From above fig. It shows that delay of proposed work is less than the reference work and operating frequency of proposed is more than the reference work.

The proposed work which achieves less area (No.of slices) was synthesized on Xilinx ISE Design Suite Spartan3E XC3S100E & Comparison on the basis such parameter is summarized in the table 2.

Table.2:Comparison of area of proposed work with reference work

Sr.No	Parameter	Proposed Work	Reference Work
1.	Number of Slices	368	2626
2.	Number of LUTs	387	5612

From above tables, it is observe that required no. of slices and no. of LUTs are 368 and 387 respectively, propagation delay is 4.04ns and operating frequency is 247.524MHz which are better results as compared to reference work.

VI.I Graphical Comparison

The graph below shows that the proposed technique offers area reduction in terms of slices as well as proposed parallel architecture achieves less delay and high operating frequency.

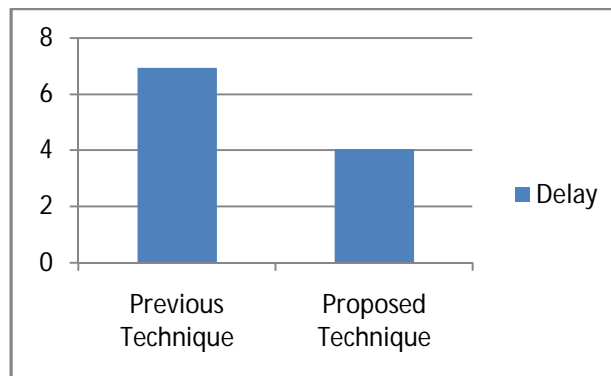


Figure.5: Graphical comparison of delay in proposed technique

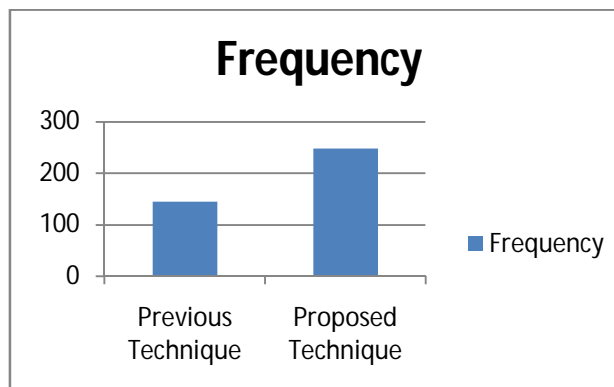


Figure.6: Graphical comparison of frequency in proposed technique



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

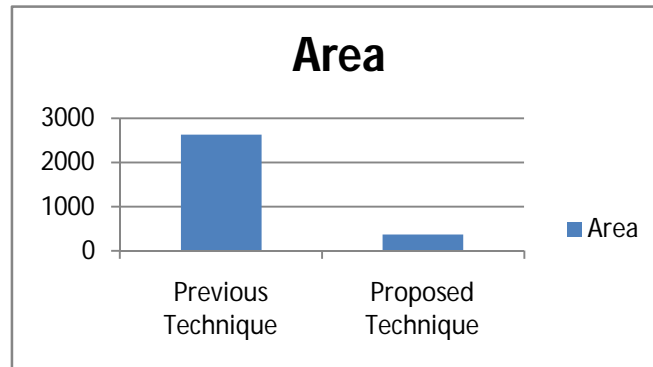


Figure.7: Graphical comparison of area in proposed technique

VII. CONCLUSION AND FUTURE WORK

Delay and area are the two key parameters of parallel CRC128 implementation. In this paper Parallel comparator based architecture is proposed for CRC128 algorithm. This proposed work is employed as substitutes for traditional approaches.

This work is helpful in increase the calculation speed of error detection and correction circuitry of software defined radio. These architecture can be executed under 247.524Mhz and we thus achieved the performance of 128 bit * 247.524Mhz = 316.83 Gbps. Compared with reference work, the proposed method consumes less area with low delay.

Future work may extend in different directions which may include the following:

1. In future, various technique will be investigated to decrease the area of the parallel CRC architecture
2. In future such an approach would be of great utility to many modern applications, for which both high-performance and low-complexity are of great importance.

REFERENCES

- [1] Neepta P. Mathew, Anith Mohan, "Matrix Code Based Error Correction For LUT Based Cyclic Redundancy Check", Global Colloquium in Recent Advancement and Effectual researches in Engineering, Science and Technology (RAEREST 2016)
- [2] Huo, Yuanhong, et al. "High performance table-based architecture for parallel CRC Calculation." Local and Metropolitan Area Network(LANMAN), 2015 IEEE International Workshop on. IEEE, 2015.
- [3] H. H. Mathukiya , and M. P. Naresh. "A Novel Approach for Parallel CRC generation for high speed application."Communication Systems and Network Technologies (CSNT), 2012 International Conference on IEEE, 2012
- [4] Yan Sun, Min Sik Kim. "A Pipelined CRC Calculation Using Lookup Tables", IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2010 proceedings.
- [5] A. Sivagami, B. Shoba and P. Raja, "An Efficient Design and Implementation of Software Radio System", International Journal of Technology and Engineering Sym, vol. 2, no. 2, Jan-March, pp. 210-216, 2011.
- [6] Wichai Pawgasame, "Evaluation of Digital Codings on the SoC -based Softwar Defined Radio for the Military Communication", Third Asian Conference on DefenceTechnology (3 ACDDT). IEEE, 2017.
- [7] Y. Sun and M. S. Kim, " A pipelined crc calculation using lookup tables", in Proceeding of IEEE Consumer Communications and Networking Conference (CCNC) , Jan. 2010.
- [8] Jaeyoung Kim, Myeongsu Kang, " A fast and energy- efficient Hamming decoder for software-defined radio using graphics processing units", Springer Science+ Business media New York 2015.
- [9] W. Lu and S. Wong, "A Fast CRC Update Implementation", IEEE Workshop on High Performance Switching and Routing ,pp. 113-120, Oct. 2003
- [10] H. H. Mathukiya , and M. P. Naresh. "A Novel Approach for Parallel CRC generation for high speed application." Communication Systems and Network Technologies (CSNT), 2012 International Conference on. IEEE, 2012.
- [11] S. Sangeeta, et al. "VLSI Implementation of Parallel CRC Using Pipelining, Unfolding and Retiming." IOSR Journal of VLSI and Signal Processing 2.5 (2013): 66-72. 2013
- [12] Harris, F., and Lowdermilk, W., "Software Defined Radio," IEEE Instrumentation and Measurement magazine, Feb. 2010.
- [13] Refaey A, Roy S, Fortier V (2011) A new approach for FEC decoding based on the BP algorithm in LTE andWiMAX systems. In: Proceedings of the 2011 12th IEEE Canadian workshop on information theory, pp 9–14, Kelowna, 17–20 May 2011.



ISSN(Online): 2320-9801
ISSN(Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

- [14] Palkovic M, Raghavan P, Li M, Dejonghe A, Van der Perre L, Catthoor F (2010) Future software-defined radio platforms and mapping flows. IEEE Signal Process Mag 27(2):22–33.
- [15] Lin Y, Lee H, Who M, Harel Y, Mahlke S, Mudge T, Chakrabarti C, Flautner K (2007) SODA: a high-performance DSP architecture for software-defined radio. IEEE Micro 27(1):114–123.
- [16] S. L. Shieh, P. N. Chen, and Y. s. Han, "Flip CRC modification for message length detection," IEEE Transaction on Communication, vol.55, no. 9, pp. 1747-1756, 2007.