FPGA Implementation of Cyclic Redundancy Check-16 Bit using Pipelining Architecture Approach

Maneesh Pawar[1], Dr. Anshuj Jain[2]

M.Tech Scholar, Dept. of ECE., SCOPE College of Engineering, Bhopal, India[1]

Associate Professor & HOD, Dept. of ECE., SCOPE College of Engineering, Bhopal, India[2]

Abstract : Cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. Fastest data transmission with high accuracy is needed rapidly. Error detection and correction is important step during data transmission. There are various algorithm is capable to perform such task. This paper proposed a design and implementation of Cyclic Redundancy check-16 bit. Further the performance improvement is done through the proposed design in terms of reduced area, reduced latency and improved throughput.

IndexTerms – CRC, Error, Correction, Speed, Power, Latency.

I. INTRODUCTION

Error detection and correction code is needed during advance data transmission scheme. CRCs are so called because the check (data verification) value is a redundancy (it expands the message without adding information) and the algorithm is based on cyclic codes. CRCs are popular because they are simple to implement in binary hardware, easy to analyze mathematically, and particularly good at detecting common errors caused by noise in transmission channels. Because the check value has a fixed length, the function that generates it is occasionally used as a hash function.

The CRC was invented by W. Wesley Peterson in 1961; the 32-bit CRC function, used in Ethernet and many other standards, is the work of several researchers and was published in 1975.

CRCs are based on the theory of cyclic error-correcting codes. The use of systematic cyclic codes, which encode messages by adding a fixed-length check value, for the purpose of error detection in communication networks, was first proposed by W. Wesley Peterson in 1961 [2]. Cyclic codes are not only simple to implement but have the benefit of being particularly well suited for the detection of burst errors: contiguous sequences of erroneous data symbols in messages. This is important because burst errors are common transmission errors in many communication channels, including magnetic and optical storage devices. Typically an n-bit CRC applied to a data block of arbitrary length will detect any single error burst not longer than n bits, and the fraction of all longer error bursts that it will detect is (1 − 2−n).

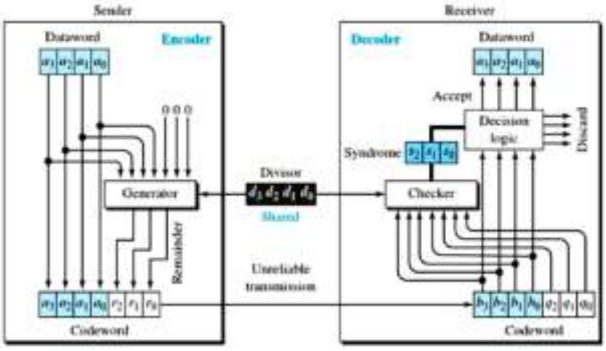


Figure 1: CRC sender and receiver block

Field-programmable gate arrays (FPGAs) have large resources of logic gates and RAM blocks to implement complex digital computations.[2] As FPGA designs employ very fast I/O rates and bidirectional data buses, it becomes a challenge to verify correct timing of valid data within setup time and hold time. Many FPGAs can be reprogrammed to implement different logic functions,[2] allowing flexible reconfigurable computing as performed in computer software.

II. BACKGROUND

N. N. Qaqos et al.,[2019]Cyclic Redundancy Check (CRC) assumes a significant job in information stockpiling, communication frameworks and networking condition fields to distinguish blunders. The speed of transmitting information with enhanced equipment utilization is the main difficulties these days. Hence, CRC calculation turns into a bottleneck for the framework implementations.[1]

W. Liu, et al.,[2019] To maintain a strategic distance from the interference among Zigbee and WiFi channels, the controller assembles information way for both of them according to the frame type to be transmitted. The main modules/segments of the controller includes the frame encapsulation, transmission, receiving and filtering, the CSMA/CA calculation, the CRC check, the programmed affirmation transmission and the programmed frame retransmission function. The controller is functionally reproduced on Xilinx xc7z020 FPGA chip using Vivado Simulator.[2]

Bajarangbali et al.,[2016] This work depicts the structure and improvement of altered CRC calculation for the equipment implementation on FPGA to meet the speed constraint for Ethernet, using the diminished lookup table calculation. This calculation can be applied for any length of information, by processing it in a square of 16 bytes one after another. The last square may have under 16 bytes. To process an input square of 16 bytes, the calculation first structures an improved table of pre-determined CRC. Corresponding to the input information, lookup from this table is done and the outcomes from the table lookup are combined by XOR operations to shape the final CRC of the input information.[3]

R. O. S.Juan, et al.,[2016] Controller Area Network (CAN) convention uses Cyclic Redundancy Check (CRC) code as a self-correcting strategy to identify and address blunders. The main goal of this calculation is to utilize an elective mistake correction conspire which is called as the Hamming code, replacing the conventional CRC code. In addition, to perhaps increase the CAN's frame pace of the framework. The bit's positions of the repetitive bits 'r' and the bit floods of the frames from the start-of-frame (SOF) to the control bit frames are determine.[4]

M. Sharma et al.,[2016] Conventional distributed arithmetic (DA) is popular in field programmable gate array (FPGA) design, and it features on-chip ROM to achieve high speed and regularity. In this paper, we describe high speed area efficient 1-D discrete wavelet transform (DWT) using 9/7 filter based new efficient distributed arithmetic (NEDA) Technique. Being area efficient architecture free of ROM, multiplication, and subtraction, NEDA can also expose the redundancy existing in the adder array consisting of entries of 0 and 1. [5]

P. Mathew, et al.,[2015] This work displays the equipment implementation of 2.4 GHz Narrowband Physical Layer for Wireless Body Area Network (WBAN) in light of IEEE 802.15.6-2012 standard. Significant building squares of PHY handset, for example, CRC, spreader, interleaver and scrambler were individually planned and integrated. To evade the inherent limitation of the information transmission and to accomplish higher unwavering quality particularly for restorative applications, BCH (63, 51, 2) encoder and decoder is integrated to the plan. [6]

Z. Shen et al.,[2014] The Dark Matter Particle Explorer (DAMPE) is being developed as a logical satellite to watch high vitality enormous beams in space. As a main finder of DAMPE, the Bismuth Germanium Oxide (BGO) calorimeter is responsible for measuring particle vitality deposition, distinguishing positrons, electrons and gamma beams from hadron foundation and providing trigger information. The satellite is intended to fly on a close earth orbit with an elevation of 500km, which sets expectations for electronic parts for the capacity to work in radiation environment.A streak based Field-Programmable Gate Array (FPGA) of Actel, ProASCI In addition to (APA) is picked as the control segment of the Front-End Electronics (Expense). To consider the radiation obstruction, SEE test was performed at the Heavy Ion Research Facility in Lanzhou (HIRFL). The chip didn't exhibit latchup to a compelling LET of 90 MeV-cm2/mg. The Blaze switches of APA, which give nonvolatile, reconfigurable interconnect programming and interface routing nets, were invulnerable to SEL. [7]

C. Chen, et al.,[2014] propose two countermeasures including power balance by using 8B/10B to encode transmitted information in the transport to neutralize power investigation, and by using CRC check to balance issue assaults on the transport. Both of the strategies have been mimicked and confirmed on FPGA board. FPGA verification results show that 8B/10B decipher quite diminishes the qualities of intensity, CRC can effectively distinguish issues injected into the transport in a proportion of 99.4%. These two countermeasures have been utilized in the bank IC card structured by our lab.[8]

III. PROPOSED METHODOLOGY

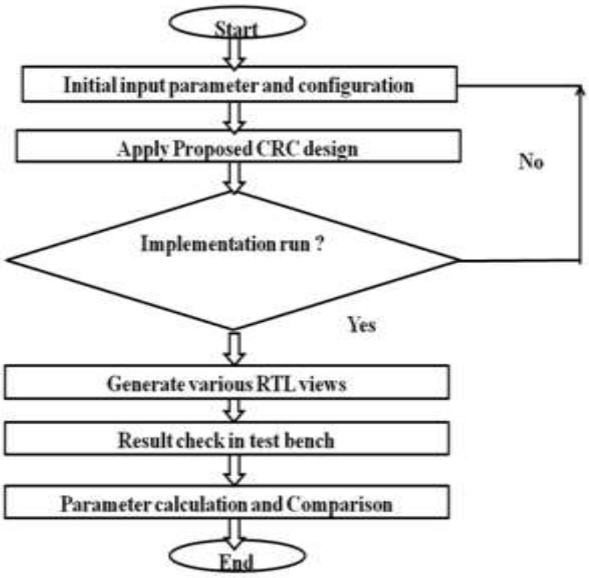

Start
Initial input parameter and configuration
Apply Proposed CRC design
No
Implementation run ?
Yes
Generate various RTL views
Result check in test bench
Parameter calculation and Comparison
End

Figure 2: Flow Chart

CRC depends on modulo-2 division, in that addition is performed by elite OR operation (i.e., the subtraction operation is indistinguishable from the addition operation, meaning 1+1=0). In CRC, an arrangement of repetitive bits (called the CRC remainder) is appended to the end of an information unit, therefore the resulting information unit turns out to be actually distinguishable by a pre-determined binary number (divisor). At its destination, the incoming information unit, which includes the information bits and equality bits, is isolated by a similar binary number (divisor). In the event that there is no remainder, that is the remainder is zero, the information unit is thought to be intact and in this manner acknowledged, while a non-zero remainder indicates that information must be dismissed and a retransmission of the information unit be mentioned. The CRC checker at the transmitter and the CRC generator at the collector function precisely the equivalent.

The divisor in the CRC generator is regularly spoken to not as a string of 1s and 0s, yet as a logarithmic polynomial. The polynomial organization is utilized for it is short and it very well may be utilized to demonstrate the idea numerically. A string of 0s and 1s can be spoken to as a polynomial with coefficients of 0 and 1, where the intensity of each term in the polynomial indicates the position of the bit and the corresponding coefficient mirrors the estimation of the bit (0 or 1). A polynomial is spoken to by removing all terms with zero coefficients. The calculation of CRC bits is as per the following:

The more bits the CRC esteem has the less is the likelihood of a collision: for CRC-8 there are just 256 diverse CRC values. This implies if the information is upset or adjusted among sender and beneficiary, there is a likelihood of 1/256 that the altered information stream has the equivalent CRC esteem as the original information stream, accordingly the error isn't distinguished. Next to different components (more on this in the hypothesis part), increasing the CRC width brings about better error protection.

This brings up the issue: What is the effect on the implementation on the off chance that we need to extend it from CRC-8 to CRC-16?

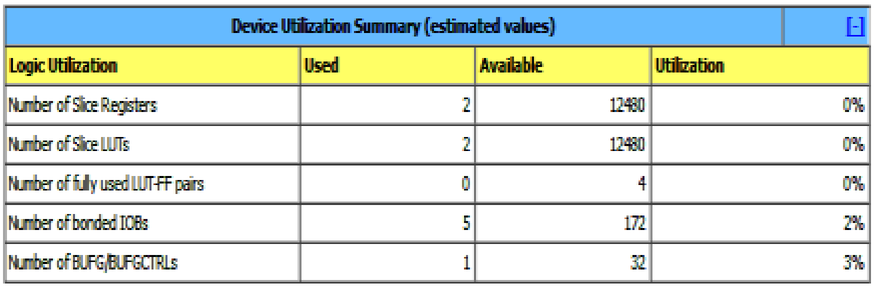
1. Clearly a CRC-16 uses a polynomial of degree 17, yet like CRC-8 the most noteworthy bit is implicitely 1. In this manner the generator polynomial just as the CRC esteem have now a 16bit information type.

2. How to 'Xor-In' the following input byte (8bit) into the CRC esteem (16bit)? The appropriate response is into the most signficant byte of the CRC.

IV. SIMULATION AND RESULT

Proposed CRC-16 design implementation and simulation is done using Xilinx 14.7 software. The behavioral modeling style is using to write the proposed algorithm. The speed is enhancing to reduce the latency and utilization area.

Table 1: Device utilization summary



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	2	12480	0%
Number of Slice LUTs	2	12480	0%
Number of fully used LUT-FF pairs	0	4	0%
Number of bonded IOBs	5	172	2%
Number of BUFG/BUFGCTRLs	1	32	3%

Table 1 is showing the using elements of proposed concepts. The total number elements using is 9 which description is showing in the used elements column.

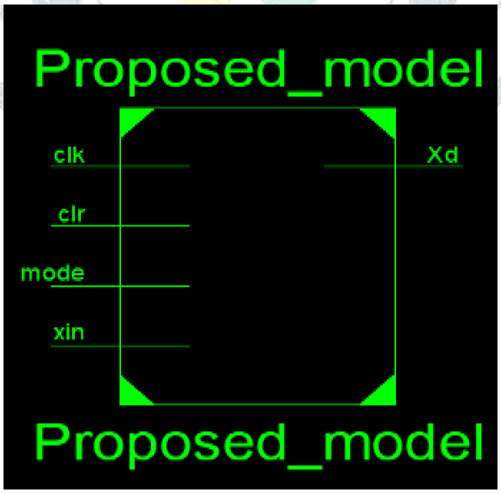


Figure 3: Proposed Model

In figure 3, showing top model of proposed design, where clock represent via clk, clear, mode and xin is showing in the input side and xd is showing the output side.

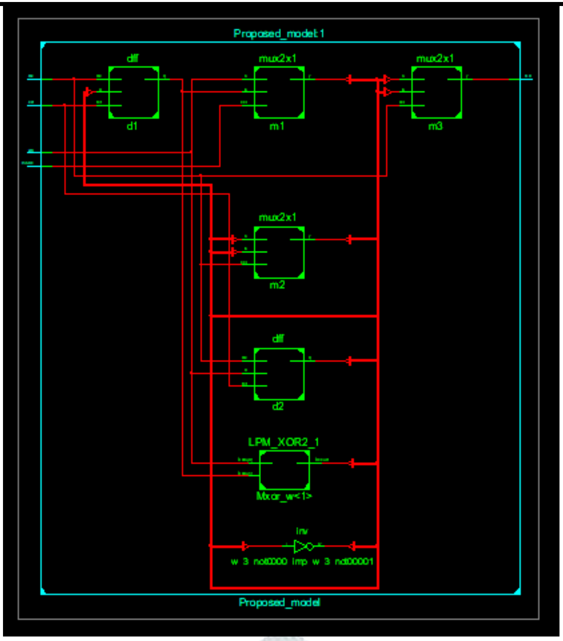
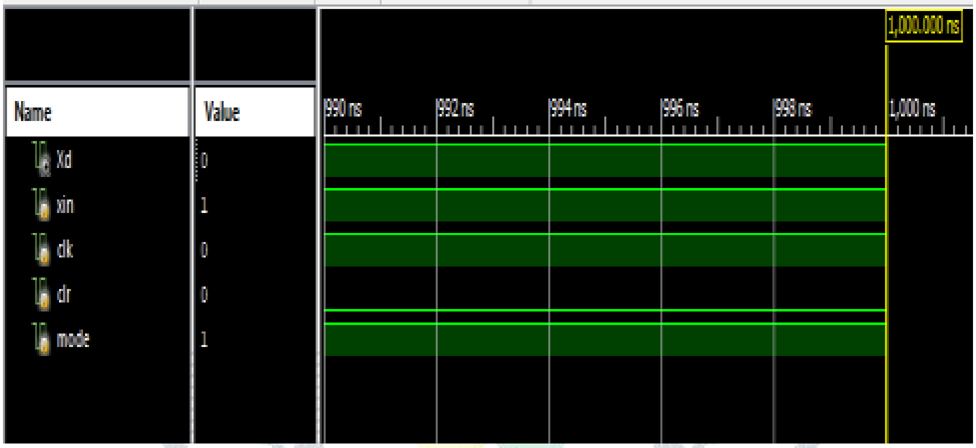


Figure 4: RTL View

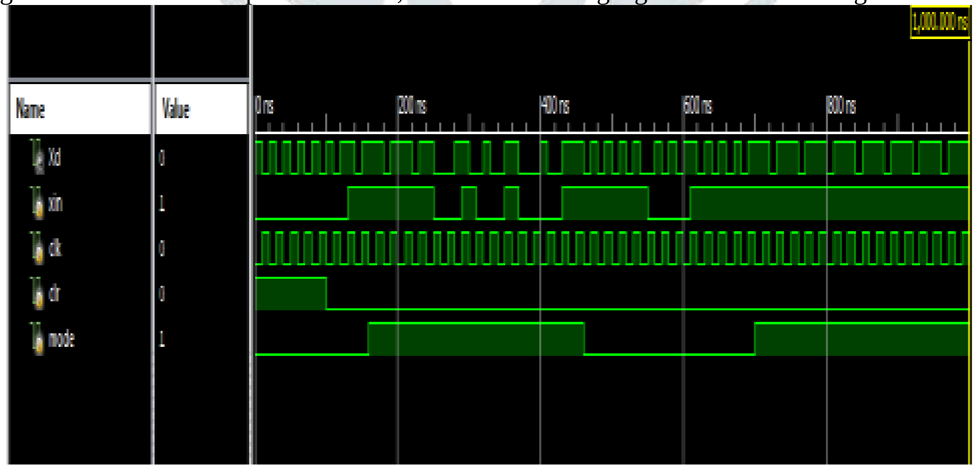
Figure 4 is showing proposed model register transfer level view. In this many of the internal wire and blocks are showing. FPGA manufacturers try to provide just enough tracks so that most designs that will fit in terms of lookup tables (LUTs) and I/Os can be routed.



Name	Value
Xd	0
xin	1
clk	0
clr	0
mode	1

Figure 5: Result in test bench-1

In figure 5, showing test bench bar for all possible value, value 1 is showing signal and 0 is showing with no signal.



Name	Value
Xd	0
xin	1
clk	0
clr	0
mode	1

Figure 6: Result in test bench-2

In figure 6, showing input, clock and output values.

Table 2: Comparison with Previous and proposed work

Sr No.	Parameters	Previous work	Proposed work
1	No of used element	124	9
2	Area (mm^2)	35	10
3	Delay (ns)	39	4.032
4	Power (mW)	1.08	0.42
5	PDP (Power delay product)	421	169
6	Throughput (Gbps)	17	18

Therefore proposed CRC gives better result in term of calculated parameters. So it can be used in high speed, low area and latency.

V. CONCLUSION

Therefore CRC has a wide scope of applications in information stockpiling gadgets and communication frameworks. CRC utilizes binary division procedure. The most well-known equipment implementation of CRC computation is the Linear Criticism Move Register (LFSR), which handle the information in sequential way and this calculation cannot an accomplish a high throughput. The proposed design executed on FPGAs, which are programmable rationale gadgets that speak to the most well-known and every now and again gadgets used to actualize a wide scope of advanced frameworks. Proposed design gives significant better result than previous design.

REFERENCES

1. N. N. Qaqos, "Optimized FPGA Implementation of the CRC Using Parallel Pipelining Architecture," *2019 International Conference on Advanced Science and Engineering (ICOASE)*, Zakho - Duhok, Iraq, 2019, pp. 46-51.
2. W. Liu, Z. Chen, G. Luo, G. Li, S. Wu and W. Zhou, "A MAC Controller Design for Zigbee and WiFi Hybrid Network Based on FPGA," *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, Tianjin, China, 2018, pp. 430-434.
3. Bajarangbali and P. A. Anand, "Design of high speed CRC algorithm for ethernet on FPGA using reduced lookup table algorithm," *2016 IEEE Annual India Conference (INDICON)*, Bangalore, 2016, pp. 1-6.
4. R. O. S. Juan, M. W. Jeong, H. W. Cha and H. S. Kim, "FPGA implementation of hamming code for increasing the frame rate of CAN communication," *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Jeju, 2016, pp. 684-687.
5. M. Sharma and P. S. Lal, “ANALYSIS OF EFFECTS OF USING 9/7 WAVELET COEFFICIENTS IN MULTI-RESOLUTION ANALYSIS”, *IJOSCIENCE*, vol. 2, no. 1, Jan. 2016. https://doi.org/10.24113/ijoscience.v2i1.68
6. Z. Shen *et al*., "Study on FPGA SEU Mitigation for the Readout Electronics of DAMPE BGO Calorimeter in Space," in *IEEE Transactions on Nuclear Science*, vol. 62, no. 3, pp. 1010-1015, June 2015.
7. P. Mathew, L. Augustine, D. Kushwaha, V. Desalphine and A. David Selvakumar, "Implementation of NB PHY transceiver of IEEE 802.15.6 WBAN on FPGA," *2014 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA)*, Bangalore, 2015, pp. 1-6.
8. C. Chen, S. You, L. Wu and X. Zhang, "A novel bus security solution for bank IC card with FPGA," *2014 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, Guilin, 2014, pp. 1-3.
9. I. M. Safarulla and K. Manilal, "Design of Soft error tolerance technique for FPGA based soft core processors," *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, Ramanathapuram, 2014,
10. T. Závodník, L. Kekely and V. Puš, "CRC based hashing in FPGA using DSP blocks," *17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, Warsaw, 2014, pp. 179-182.
11. U. Quadri, P. Rangaree and G. M. Asutkar, "FPGA implementation of an emulator for Wireless Sensor Node with Pt100 temperature sensor," *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*, Xi'an, 2013, pp. 1-5.
12. A. Akagic and H. Amano, "Performance analysis of fully-adaptable CRC accelerators on an FPGA," *22nd International Conference on Field Programmable Logic and Applications (FPL)*, Oslo, 2012, pp. 575-578.