The CSI Journal on
Computer Science and Engineering
Vol. 14, No. 1, 2016
Pages 19-22
Short Paper

Novel Quaternary Operator for Cyclic Redundancy Check on Xilinx FPGAs

Mahya Sam Daliri Saeed Sam Daliri Ali Bozorgmehr Keivan Navi

Faculty of Computer Engineering and Science, Shahid Beheshti University, Tehran, Iran

Abstract

Cyclic Redundancy Check can be used in computer networks and data storage devices to provide fruitful error detection ability. A novel quaternary operator for CRC application is presented in this paper. The CRC circuit using the proposed operator and also SUM operator are described by VHDL and then Xilinx FPGA is used to synthesize and to perform place and route. Furthermore implementation of CRC circuit on FPGAs using the proposed operator outperforms the same circuit using SUM operator in terms of speed, power and area-efficiency.

Keywords: CRC, MVL, LFSR, Error Detection, Quaternary Operator, FPGA.

1. Introduction

In VLSI design, the interconnections occupy a large amount of chip area and lead to problems and restrictions in placement and routing of logic elements which increase the number of connections inside and outside of the binary logic circuits [1, 2]. One of the approaches to solve such problems and reduce the number of interconnections on a given chip is using higher radices and subsequently Multiple-Valued Logic (MVL) instead of binary logic. Since there has been a promising trend on the way to Multiple-Valued Logic, the design and implementation of MVL circuits are needed considerably. Moreover, the quaternary logic consisting of four digits {0, 1, 2, 3} benefits from simple conversion between quaternary and binary signals and has the advantage of easier communication with binary logic circuits therefore VLSI designers using the MVL have focused on the quaternary systems [3].

One of the most common circuits used in data communication is Cyclic Redundancy Check (CRC) circuit. CRC is an error detection code that is mostly used in computer networks and data storage devices. It is based on cyclic redundancy code known as polynomial manipulations using modulo arithmetic. A linear feedback shift register (LFSR), comprising flip-flops and XOR (Exclusive-OR) gates, can be used to hardware implementation of CRC computation. The LFSR can be denoted by a generator polynomial of variable X in Galois field configuration named G(X) whose coefficients are either zero or one [4]. The number of used flip-flops is equal to the order of G(X) named m. The flip-flops are numbered from “0” to “m-1”. The coefficient values of G(X) describe the location of feedback taps for the LFSR so that if a XOR gate exists in the input of any flip-flop, the number of that flip-flop will be appeared as the power of the G(X).The output of LFSR which is the output of the last (MSB) flip-flop is fed back into the input of the first (LSB) flip-flop.

For this reason, there will be always “0” and “m-1” powers in G(X). In transmitter side each flip-flop of the LFSR is set to zero at the initial stage. The input data is shifted into the LFSR bit-by-bit per clock cycle [4]. This will be continued until the last bit of the input data stream is processed. The resulting values of the flip-flops become the CRC and add to the end of the original input data stream. This new data stream is sent to the receiver. In receiver side, the same LFSR is used and the same procedure of the transmitter side will be done. If there is no error in transmission, the final value of the flip-flops should be zero

[5]. As demonstrated, XOR is the most commonly used function in the LFSR design. On the other hand, to design CRC computation circuit in MVL systems, MVL XOR function is required [6]. The properties of binary XOR include self-inverse (Eq.1), identity (Eq.2), commutativity (Eq.3), and associativity (Eq.4). These properties can be applied bitwise to a vector of bits. However, XOR may show unusual behavior in MVL compared to binary logic. [2].

$$a \oplus a = 0 \tag{1}$$

$$a \oplus 0 = a \tag{2}$$

$$a \oplus b = b \oplus a \tag{3}$$

$$a \oplus (b \oplus c) = (a \oplus b) \oplus c \tag{4}$$

In this paper a novel quaternary XOR function with emphasis on the use of CRC algorithm is presented. There have been many implementations for quaternary circuits, such as full adder [3], memory [7], Arithmetic Logic Unit [8] and flip-flop [9]. But no quaternary operator suitable for CRC circuit has been reported in the literatures. The rest of the paper is organized as follows. Section 2 presents the proposed quaternary operator. Also, a brief discussion about conventional quaternary operator (SUM) is given in this section. Synthesis, implementation and comparison of CRC circuit on FPGAs using the proposed operator and SUM are provided in Section 3 and finally, Section 4 concludes the paper.

2. The Proposed CRC Quaternary Operator

Depending on the considering application, one or more features of binary XOR can be used for XOR in MVL like quaternary logic. The fundamental properties of XOR function for performing in CRC circuit are specified in Eq.1 and Eq. 2 [10]. Earlier, based on these two essential characteristics, the two ternary operators were provided in [6] and [10] for CRC computations in ternary logic but there is no quaternary operator using in CRC circuit.

Although the common explanation of MVL XOR is module-Radix addition called SUM operator but this operator in module-4 is not suitable for CRC computation because it does not satisfy the Eq.1. However, it is practicable to use this operator due to the fact that three successive SUM operators provide the essential condition as shown in table 1. The operator SUM is applied three times at the beginning of LFSR in both transmitter and receiver sides for correct CRC calculation. If the length of the input data is longer, then this replication will have adverse effect in both sides.

Table 1. The essential property for data verification provided by 3-times the operator SUM

a	b=SUM (a,a)	c=SUM (b,a)	SUM (c,a)
0	0	0	0
1	2	3	0
2	0	2	0
3	2	1	0

The general structure of quaternary CRC circuit is shown in figure 1. The XOR used in this structure is a quaternary XOR. In the transmitter side, SW1 switches to set zero in each flip-flop at the initial stage. The second switch (SW2) select the incoming data. Also SW2 selects the final values of the flip-flops (CRC code) which should be sent in the reverse direction to the receiver side. In the receiver side (Figure 1 b), the switches turn to conduct the final value of each flip-flop to an OR gate to detect the error. We propose a CRC quaternary operator (QuaCOp) which is indicated in Eq. 5.

$$QuaCOp(a,b) = \begin{cases} (a-b) \bmod 4 & if\,(a \neq 0) \\ b & else \end{cases} \tag{5}$$

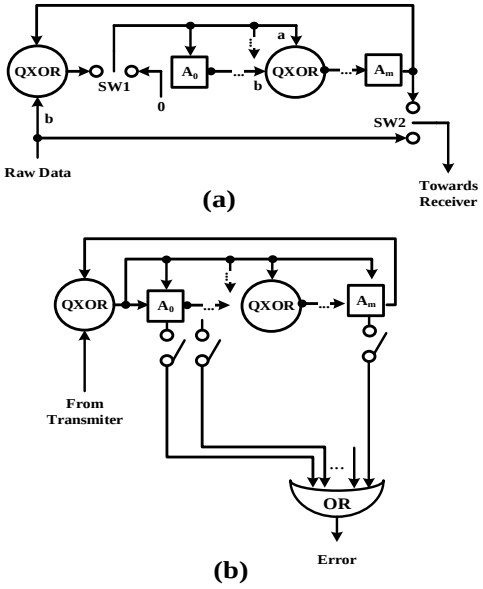

Figure 1. Quaternary CRC circuits (the general structure), (a) In the transmitter side, (b) In the receiver side

Table 2. The truth table of the quaternary operators

a	b	QuaCOp	SUM
0	0	0	0
0	1	1	1
0	2	2	2
0	3	3	3
1	0	1	1
1	1	0	2
1	2	3	3
1	3	2	0
2	0	2	2
2	1	1	3
2	2	0	0
2	3	3	1
3	0	3	3
3	1	2	0
3	2	1	1
3	3	0	2

The truth tables of the proposed operator and SUM are shown in table 2. The proposed operator satisfies the self-inverse and identity conditions (Eq.1 and Eq.2) which are necessary to fulfil in CRC circuit. As shown in figure 1 a, the variable b (second variable of XOR operator) is intended as the input data for LFSR and as the second input of each

quaternary XOR operator used in the LFSR. As depicted in table 1, if the first variable of QuaCOpis constant, this operator returns distinct values (QuaCOp(a, 0) ≠QuaCOp (a, 1) ≠QuaCOp (a, 2) ≠QuaCOp (a, 3)).

3. Synthesis, Implementation and Discussions

The hardware implementation of a typical CRC circuit on FPGAs using the proposed operator and SUM (Figure 2) are written with Hardware Description Language (VHDL), and then are synthesized using Xilinx Integrated Software Environment (ISE) design suite 14.6. The simulation helps to verify the designs and the synthesis report gives the speed and area of the designs. Finally, the VLSI implemented designs are targeted to the FPGA device Spartan6 family, part XC6SLX4, package TQG144 and captured some parameters such as the real time speed and area of the designs.

The CRC circuits in the receiver side using QuaCOp and SUM operators are also shown in figure 3. The input data for each operator includes the original data + transmitter CRC code. The transmitter CRC code is the final value of flip-flop in reversed direction (D_2 D_0 D_1).

The analysis are done at room temperature, 1.2V supply voltage and speed grade 3. The simulation results are shown in table 2. Various key performance metrics such as number of slice registers, number of slice LUTs displaying the area-efficiency, total delay, maximum frequency and total power consumption are estimated for the circuits. Simulation results determine that using the proposed operator for CRC calculation leads to better speed-performance, lower power and higher area-efficiency compared to SUM operator.

Table 3. synthesis report of the CRC circuits

	CRC using QuaCOp	**CRC using SUM**
No. of Slice Registers	10	14
No. of Slice LUTs	8	12
Total memory usage	254796KB	255244KB
Total Delay	3.752ns	3.84ns
Maximum frequency	669.68 MHz	655.50 MHZ
Total Power	14 mW	15 mW

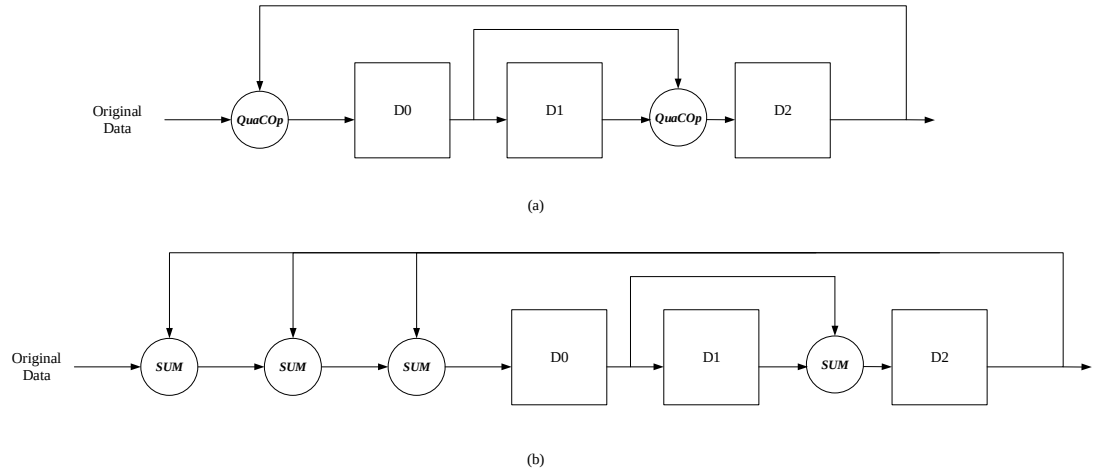

Figure 2. Typical CRC circuits in transmitter side using (a) QuaCOp, (b) SUM

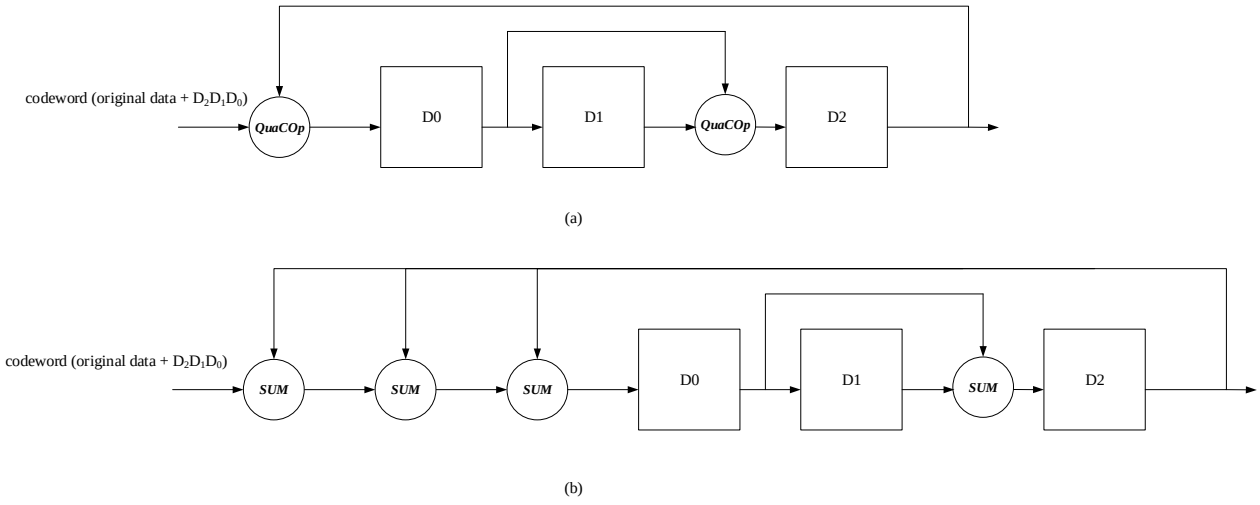

Figure 3. Typical CRC circuits in receiver side using (a) QuaCOp, (b) SUM

4. Conclusion

CRC is a well-known error detection code employed in many different communication protocols. Hitherto no quaternary CRC operator has been reported in the literatures. This paper presents a new quaternary operator for CRC circuit. The implementation of CRC circuit on FPGAs using the proposed operator and SUM operator have been simulated and synthesized on Xilinx ISE 14.6 platform and their parameters are captured. The operator SUM should be used 3 times in the CRC circuit to work correctly. Simulation results show the superiority of using the proposed operator for CRC calculation in terms of speed, power and area-efficiency compared to SUM operator.

References

[1] E. Dubrova, "Multiple-Valued Logic in VLSI: Challenges and Opportunities," *in Proc. NORCHIP Conference*, Oslo, Norway, Nov. 1999, pp. 340-350.

[2] S. L. Hurst, "Multiple-valued logic - Its status and its future," IEEE Transactions on Computers, vol. 33, no. 12, pp. 1160-1179, Dec. 1984.

[3] Sharifi, Fazel, M. H. Moaiyeri, K. Navi, and N. Bagherzadeh "Quaternary full adder cells based on carbon nanotube FETs," *Journal of Computational Electronics* 14, no. 3 (2015): 762-772.

[4] E. Dubrova, and S. Mansouri, "A BDD-based approach to constructing LFSRs for parallel CRC encoding," *in Proc. of International Symposium on Multiple-Valued Logic*, pp. 128–133, 2012.

[5] M. Grymel, and S. B. Furber, "A novel programmable parallel CRC circuit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 10, pp. 1898–1902, 2011.

[6] IBM Tech. Disc. Bulletin, "Cyclic Redundancy Checking For Multiple Valued Logic," vol. 34, no. 11, Apr. 1992.

[7] S. Zhang, and et al., "Quaternary memory based on magnetic skyrmion," 2015 *IEEE Magnetics Conference (INTERMAG)*. IEEE, 2015.

[8] A. N, Nagamani, and S. Nishchai, "Quaternary High Performance Arithmetic Logic Unit Design," *In Digital System Design (DSD)*, 2011 14th Euromicro Conference on, pp. 148-153. IEEE, 2011.

[9] A. Mochizuki, H. Shirahama, and T. Hanyu, "Design of a low-power quaternary flip-flop based on dynamic differential logic," *IEICE transactions on electronics* 89, no. 11 (2006): 1591-1597.

[10] M. S. Daliri, R. F. Mirzaee, K. Navi, and N. Bagherzadeh, "Ternary cyclic redundancy check by a new hardware-friendly ternary operator," *Microelectronics Journal* 54 (2016): 126-137.

Mahya Sam Daliri is currently working toward the PhD degree in computer hardware engineering at Shahid Beheshti University, Tehran, Iran. She is also a member of the Nanotechnology and Quantum Computing Lab of Shahid Beheshti University. Her research interests mainly focus on VLSI design, Multiple-Valued Logic, computer arithmetic, signal processing and nano electronics circuitry with an emphasis on molecular, CNFET, QCA and low-power, and high-performance logic circuit design.
E-mail: m_sam@sbu.ac.ir

Saeed Sam Daliri received his BS and MS in computer hardware engineering from Babol Noshirvani University of Technology, Babol, and University of Mohaghegh Ardabili, Ardabi, Iran, respectively. He is also a member of the Nanotechnology and Quantum Computing Lab of Shahid Beheshti University. His main area of research interests are VLSI circuit design, low power arithmetic, and nano electronics circuitry.
E-mail: samdaliri.s@student.uma.ac.ir

Ali Bozorgmehr is currently working toward the PhD degree in computer hardware engineering at Shahid Beheshti University, Tehran, Iran. He is also a member of the Nanotechnology and Quantum Computing Lab of Shahid Beheshti University. His research interests include cryptography, signal processing, security computing, nanoelectronic circuit design, low power arithmetic, fuzzy systems, and approximate computing.
E-mail: a_bozorgmehr@sbu.ac.ir

Keivan Navi received the B.Sc. and M.Sc. degrees in electrical engineering (hardware engineering) from Shahid Beheshti University, Tehran, Iran, in 1987 and Sharif University of Technology, Tehran, Iran, in 1990, respectively. He also received the Ph.D. degree in computer architecture from Paris XI University, Paris, France, in 1995. He is currently a Full Professor in Faculty of Computer Science and Engineering of Shahid Beheshti University and also a senior member of IEEE. He is in charge of the Nanotechnology and Quantum Computing Laboratory (NQC Lab.). His research interests include Nanoelectronics with emphasis on CNFET, QCA and SET, computer arithmetic, interconnection network design, quantum computing, cryptography, Multiple-Valued Logic, fuzzy systems, and approximate computing.
E-mail: navi@sbu.ac.ir

Paper Handling Data:

Submitted: 09.11.2016
Received in revised form: 11.12.2016
Accepted: 22.12.2016
Corresponding author: Dr. Keivan Navi,
Faculty of Computer Engineering and Science, Shahid Beheshti University, Tehran, Iran.