J. Basic. Appl. Sci. Res., 6(5)34-39, 2016
© 2016, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

A Novel Approach for Error Detection Using Additive Redundancy Check

[1]Shibli Nisar, [2]M.Asadullah, [3]Gulfam Nasir

[1, 2, 3] Department of Electrical Engineering, National University of Computer and Emerging Sciences Peshawar, Pakistan

Received: January 16, 2016
Accepted: April 23, 2016

ABSTRACT

In a data communication system binary data transmit from one node to another node, during transmission data can be corrupted by transmission impairments like attenuations, distortion and noise. For analog communication signal can be corrupted by external and internal noise of system and also by other transmission impairments. Digital signal can be effected by single bit error and burst error, in these errors binary 1 is changed to binary 0 and vice versa. At receiver it is check either data has corrupted or not by using error detection techniques, if data has corrupted then various error correction techniques are used to get original data. In this research work, Additive redundancy check (ARC) technique is proposed to detect errors. The suggested technique is based on basic addition. In this technique binary data is arrange in square pattern by keeping number of rows and columns equal, and then basic addition operation is apply on it. The proposed technique can easily detect those errors which other techniques fail to detect, so this technique is more robust than other existing techniques.
KEYWORDS: Error detection, Single bit error, Burst error, 2D parity check, Checksum.

1. INTRODUCTION

When binary data is transmitted in communication systems, it can be corrupted due to transmission impairments. Data corruption means occurrence of errors e.g. binary 0 changed to binary 1 and vice versa. There are two main types of error 1st is single bit error and 2nd is burst error or multiple bits error. In single bit error only one bit effected. Rest of all the binary data remains same. But in burst error two or multiple bits change it corrupt consecutive sequence of N bits. At receiver site error detection method are used as checking function, if data is corrected then accept it else reject it. Error detection can be perform by adding some redundancy bits at the end of data to be transmit, at the receiver site binary data decoded and checked either error occur or not [1,2]. There are many error detection methods available which are used to detect error by appending some extra redundancy bits with data bits. Some of these famous methods are VRC, LRC, CRC, checksum and 2D parity Check [3-8]. In VRC after counting number of 1's is even or odd extra bits are appended at the end of binary data, it can only detect burst error if number of changed bits are ODD. While LRC can detect single bit error and burst error but If in a same column 0 is changed to 1 and vice versa LRC can't detect it. In checksum 1st complement is taken after adding all binary data, then appended this complement at the end of binary data. Checksum can't detect error if two bits swap in a column. In 2D parity check the parity of each row and columns is calculated by counting number of 1's then transmit row and column parity with data bits, this method takes more redundancy bits.

In this research work, an effective method is proposed that is simply based on binary addition. It can easily detect single bit error and burst error. In this method less redundancy bits can cover more binary data bits, it can easily detect those errors which other methods failed to detect. The performance of proposed method is also compared with other existing methods in this research work.

2. METHODOLOGY

In this section all the steps are described which are involved in transmitter and receiver site.

2.1 Transmitter site

Let us consider the length of binary data is n^2 bits, if number of data bits are not squared value then append 0's at most significant bit to make it squared. Then make square (n × n) pattern and name it **A** after that take transpose of that pattern and name it $\mathbf{A}'$.
Now find ARC by following 3 steps.

***Corresponding Author:** Shibli Nisar, Department of Electrical Engineering, National University of Computer and Emerging Sciences, Peshawar, Pakistan

1. Add all the columns of pattern **A** and name it **Sum A**.
2. Add all the columns of pattern **A′** and name it **Sum A′.**
3. Now to get ARC add **Sum A** with **Sum A′.**

If carry occur then carry bits will be discard.

After calculating ARC bits by above 3 steps, transmit data bits along ARC bits. Transmitter and receiver both should have same mechanism Block diagram for Additive Redundancy Check (ARC) at Transmitter site is shown below in figure 1.

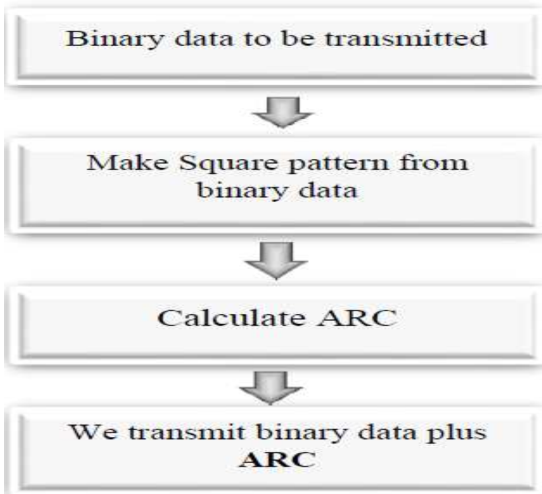

Figure 1. ARC method at Transmitter site

2.2 Receiver site

At receiver site binary data is received along redundancy bits (ARC). To check either received data is correct or not again calculate ARC by following same steps as followed at transmitter site. After calculating ARC at receiver site compare it with Transmitter site ARC, If receiver site ARC is equal to Transmitter site ARC, it means there is no error occur and data is received correctly else there is an error occurred and received data is corrupt. This method can easily detect most of error which other methods fails to detect. Block diagram for Additive Redundancy Check at Receiver site is shown below in figure 2.

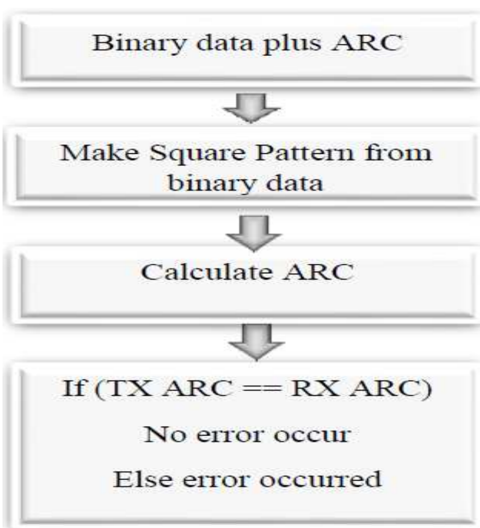

Figure 2. ARC method at receiver site

3. RESULTS

To check performance and accuracy of this method, single bit and burst error were introduced to check either this method detect it or not. Some examples are given below. At transmitter site 16 bit binary data was arranged it into square (4 × 4) pattern and calculated its ARC, example of ARC is shown below in figure 3.

3.1 Single bit error detection

It is assumed that during transmission error occurred at 1st bit position. To check that during transmission error occurred or not, again calculated ARC at receiver site. Calculation of ARC at receiver site is shown in figure 4. As shown transmitter site ARC is not equal to receiver site ARC it proved that error has been occurred during transmission.

$$1011 \neq 0111$$

Graphical view of single error detection using ARC is shown below in figure 5.

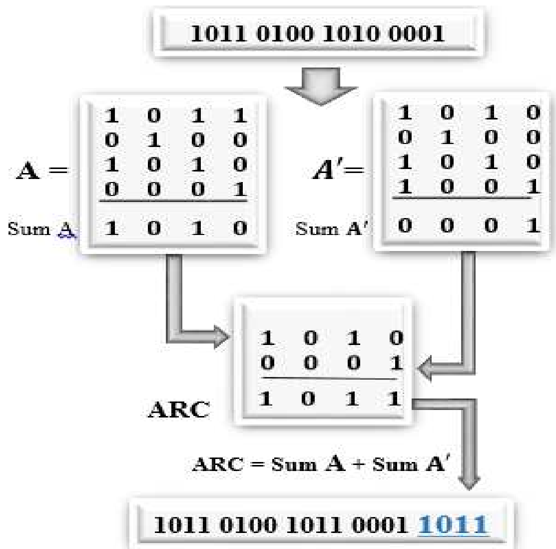

Figure 3. Example of ARC at transmitter site

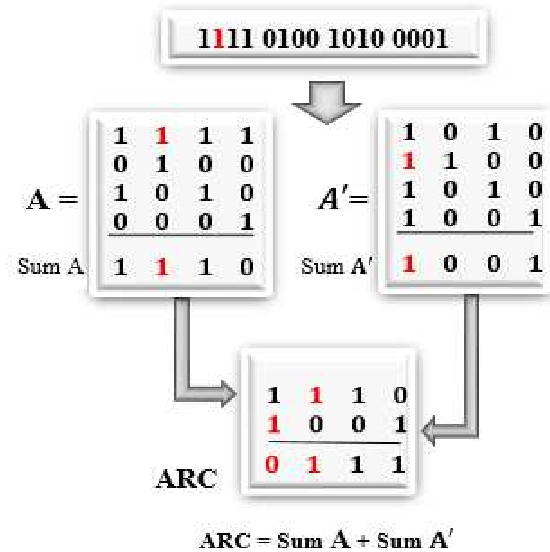

Figure 4. ARC for Single bit error detection

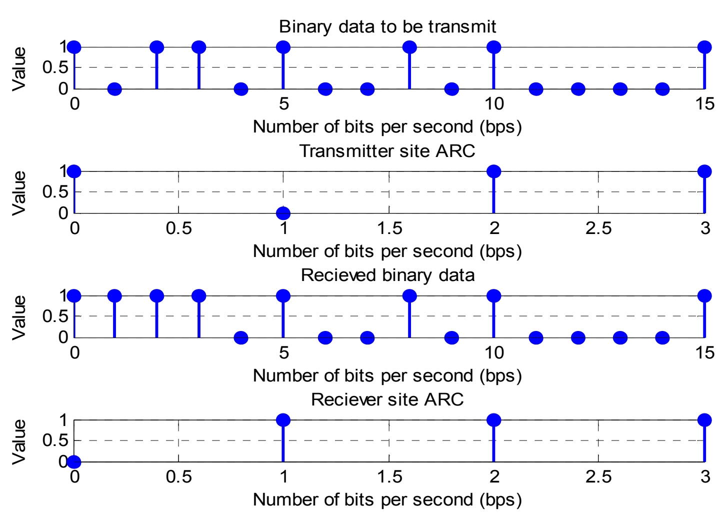

Figure 5. Graphical view of ARC method for single bit error detection

In above graph it is shown transmitter and receiver site ARC are not equal. We can also see that error is occurred at 1st position, at transmitter site there is 0 at 1st bit position but at receiver site there is 1 as shown in graph. It shows occurrence of error.

3.2 Burst error detection

It is assumed during transmission error occur at 3rd, 10th, 13th, 14th and 15th position. Calculation is shown below in figure 6.

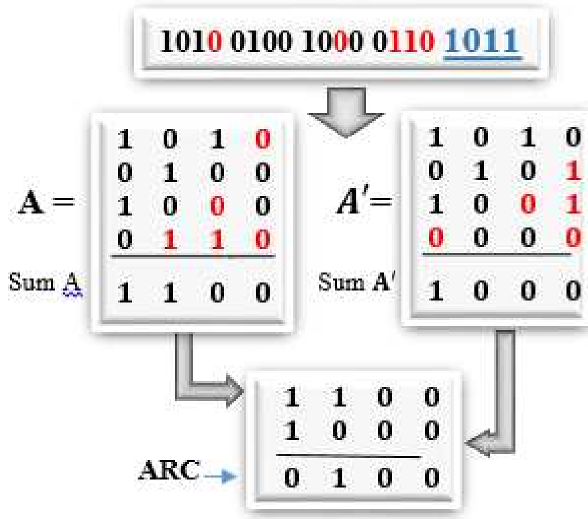

Figure 6. Example of ARC for burst error detection

As shown in above figure after passing through channel at receiver site when ARC calculated it was not equal to transmitter site ARC it shows an error occurred.

$$1011 \neq 0100$$

Graphical view of binary data at transmitter and receiver site is shown below in figure 7.

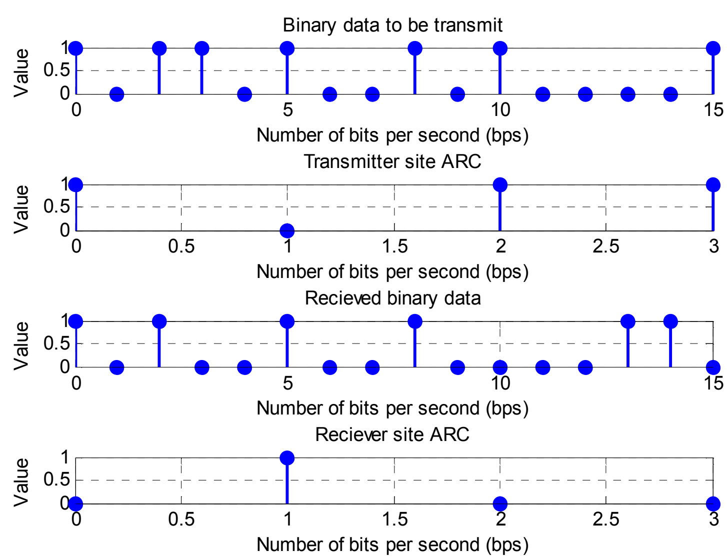

Figure 7. Graphical view of ARC method for burst error detection

From the above graph it can observe transmitter and receiver site ARC is not equal. It shows occurrence of error.

4. Analysis

If number of data bits are not squared value then append 0's at most significant bit to make it squared.
Number of data bits= N^2
Number of ARC bits= N
Total no. of (data bits + ARC bits) = $N + N^2$
Overhead =$\frac{No.\ of\ ARC\ bits}{No.of\ data\ bits} = \frac{N}{N^2}$
Code rate =$\frac{No.\ of\ data\ bits}{No.of\ (\text{data bits + ARC bits})} = \frac{N^2}{N+N^2}$

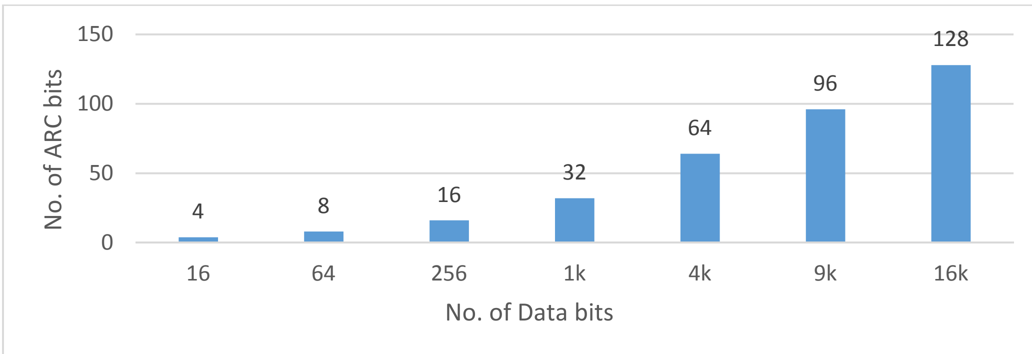

Figure 8. Analysis of No. of Data bits versus no. of ARC bits

5. DISCUSSION

Comparison of ARC method with other existing methods shows that it can easily detect all those errors which other methods cannot detect. VRC used to detect single bit error it can only detected burst error when total number of changed bits are ODD. When number of bits changed are even VRC cannot detect error. LRC can detect single bit and burst error. If in a same column 0 is changed to 1 and vice versa LRC can't detect it. Checksum is very effective error detection method, if bit inversion in one segment is balanced by bit inversion in another corresponding segment then Checksum fails to detect error. In 2D parity check, parity of individual column and row is calculated. For a long binary data it takes more redundant bits. If even number of 0's swap with even number of 1's and vice versa, then 2D parity check fails to detect errors. While Additive Redundancy Check (ARC) is based on basic arithmetic operation (Addition), in ARC less redundant bits that covers more data. It is more effective than other methods. It can easily detect those errors which other methods fail to detect. Comparison of ARC method with other error detection methods is shown below in table 1.

Table 1. Comparison table of ARC with other methods

Method name	Original data	Corrupted data	Error detection
VRC	01100111	01100100	No
LRC	01100111 11010110	00100110 11010111	No
Checksum	00100111 11010110	00100110 11010111	No
2D parity check	00100111 11010110	11100111 00010110	No
ARC	00100111 11010110	11100110 00010111	Yes

As shown in above table ARC method detects those error which other methods unable to detect, that's why ARC method is more robust than other existing methods.

6. Shortcoming

The probability of failure of this method is very rare. This technique fail to detect error when 2 bits inversion is take place in one row balanced by opposite 2 bits inversion in corresponding row. (For example, in a row two bits 01 changes to 10 and in other corresponding row two bits 10 changes to 01. In this case ARC will remain same and fail to detect error). Example is shown in figure 9.

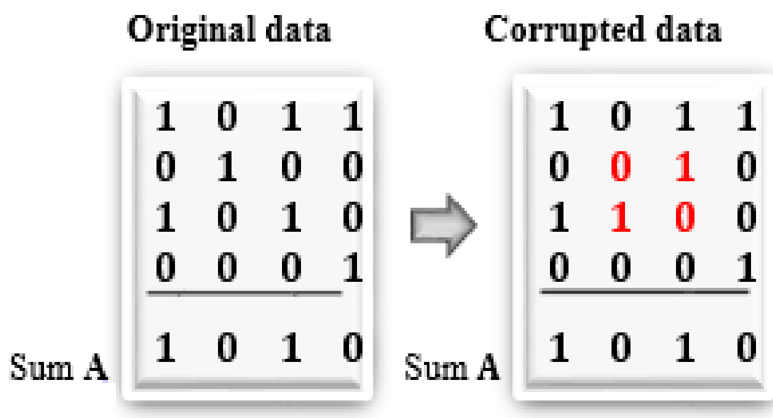

Figure 9. Block diagram of ARC shortcoming example

As shown in above example when in a row **10** changes to **01** and same time in other row **01** changes to **10** then this method fails to detect it because the sum remains same and it has no effect on ARC. In future work we will try to overcome this shortcoming and try to make its results better and more robust.

7. Conclusion

There are different error detection methods every method has its own advantages and disadvantages as already discussed. Additive redundancy check (ARC) method is effective and has better performance as compare to other methods. This method has ability to detect all those errors which undetected by other methods. ARC method is easier because it is based on basic Addition. It takes less redundancy bits to cover more data bits which mean ARC method is more fast and efficient than other methods, this is one of its advantage over all existing error detection methods. Shortcoming of this method is already discussed in previous sections and in future we will try to resolve it.

REFERENCES

[1] "Data Communication and Networking" 4th edition by Behrouz A. Forouzan (Chapter 10th page 267)
[2] "Modern Digital and Analog Communication Systems" 3rd edition by B.P Lathi (Chapter 16th page 728)
[3] "Logic and Computer Design Fundamentals" 2nd edition (Chapter 2nd page 74)
[4] "Data and Computer Communication" 8th edition by William Stalling
[5] "Mathematical Challenge April 2013 Error-correcting codes" (PDF). Swiss Quant Group Leadership Team. April 2013.
[6] https://www.researchgate.net/publication/236202666_Simple-Error-Detection-Method
[7] MacKay, David J.C. (September 2003). In-formation Theory, Inference and Learning Algorithms. Cambridge: Cambridge University Press. ISBN 0-521-64298-1.
[8] Moon, Todd K. (2005). Error Correction Coding. New Jersey: John Wiley & Sons. ISBN 978-0-471-64800-0.
[9] Peterson, W. W. and Brown, D. T. (January 1961). "Cyclic Codes for Error Detection". Proceedings of the IRE 49 (1): 228–235.doi:10.1109/JRPROC.1961.287814.