© 2018 IJSRST | Volume 4 | Issue 7 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Error Corrector Mechanism Applied to Resolve Detected Error in Data Communication

Nirvikar Sharan Katiyar[1], Dr. Pradeep Mittal[2]
[1]Ph.D Scholar, Kurukshetra University Kurukshetra, Kurukshetra, Haryana, India
[2]Kurukshetra University Kurukshetra, Kurukshetra, Haryana, India

ABSTRACT
In digital systems, the analog signals will change into digital sequence. This sequence of bits is called as "Data stream". The objective of this paper that the change in position of single bit also leads to major error in data output. In this paper we find errors and we use error detection and correction techniques to get the exact or approximate output.

Keywords : Types of errors, Error Detecting Codes, Error Detection and Correction Techniques, Parity check, checksum, polynomials.

I. INTRODUCTION

Networks must be able to transfer data from one device to another with complete accuracy. Data can be corrupted during transmission. For reliable communication, errors must be detected and corrected. Error detection and correction are implemented either at the data link layer or the transport layer of the OSI model. An error is a situation when the output information does not match the input information. During transmission, digital signals suffer from noise that may introduce errors in binary bits traveling from one system to another. This means that 0 bit can change to 1 or 1 bit can change to 0.. In digital communication system errors are transferred from one communication system to another with data. If these errors are not detected and corrected, the data will be lost. For effective communication, data must be transferred with high accuracy. This can be achieved by first detecting errors and then correcting them. Error detection is the process of detecting errors in the data transmitted in a communication system from the transmitter to the receiver. We use some redundancy codes to detect these errors, connecting them to the data while it is transmitted from the source (transmitter). These codes are called "Error detecting codes ".

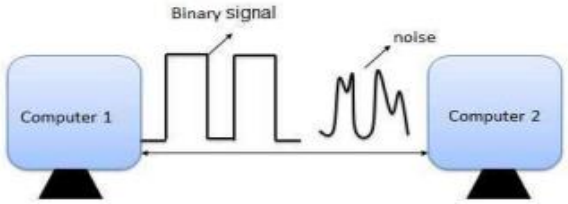
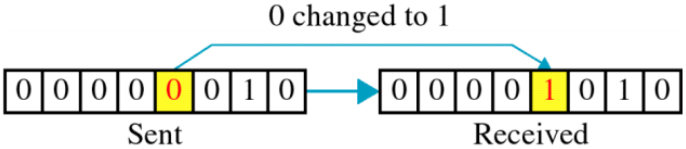


Fig 1: Data Transmission

Types of Errors

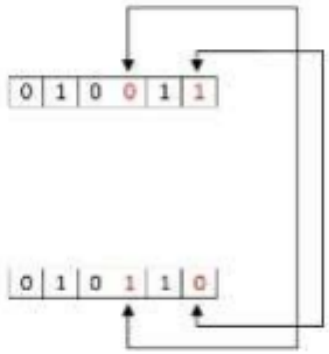
In a data sequence, if 1 is changed to zero or 0 is changed to 1, it is called "Bit error". There are generally 3 types of errors occur in data transmission from transmitter to receiver. They are Single Bit Data Errors • Multiple Bit Data Errors • Burst Errors

a. Single Bit Data Errors



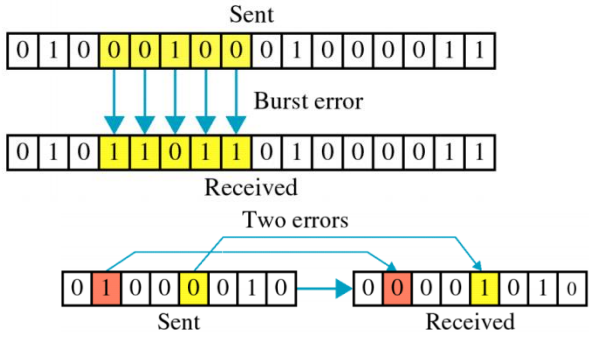
Single bit errors are the least likely type of errors in serial data transmission because the noise must have a very short duration which is very rare. However this kind of errors can happen in parallel transmission. Example: If data is sent at 1Mbps then each bit lasts only 1/1,000,000 sec. or 1 μs. For a single-bit error to occur, the noise must have duration of only 1 μs, which is very rare. The change in one bit in the whole data sequence, is called "Single bit error". Occurrence of single bit error is very rare in serial communication system. This type of error occurs only in parallel communication system, as data is transferred bit wise in single line, there is chance that single line to be noisy.

Multiple Bit Data Errors If there is change in two or more bits of data sequence of transmitter to receiver, it is called "Multiple bit error". This type of error occurs in both serial type and parallel type data communication networks.



Burst Errors The change of set of bits in data sequence is called "Burst error". The burst error is calculated in from the first bit change to last bit change

Burst Error

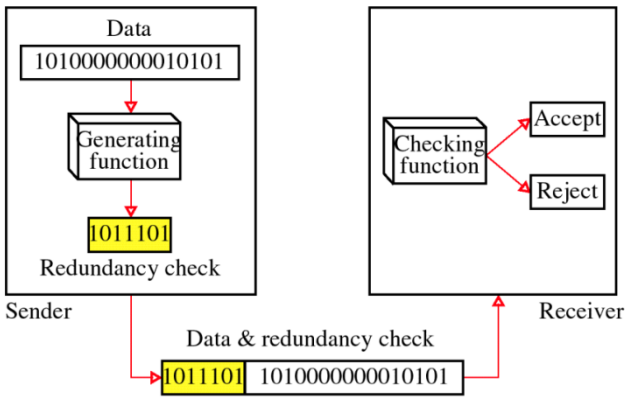


The term burst error means that two or more bits in the data unit have changed from 1 to 0 or from 0 to 1. Burst errors do not necessarily mean that the errors occur in consecutive bits, the length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted. Burst error is most likely to happen in serial transmission since the duration of noise is normally longer than the duration of a bit. The number of bits affected depends on the data rate and duration of noise. Example: If data is sent at rate = 1Kbps then a noise of 1/100 sec can affect 10 bits.(1/100*1000) If same data is sent at rate = 1Mbps then a noise of 1/100 sec can affect 10,000 bits.(1/100*10^6). Here we identify the error form fourth bit to 6th bit. The numbers between 4th and 6th bits are also considered as error. These set of bits are called "Burst error". These burst bits changes from transmitter to receiver, which may cause a major error in data sequence. This type of errors occurs in serial communication and they are difficult to solve.

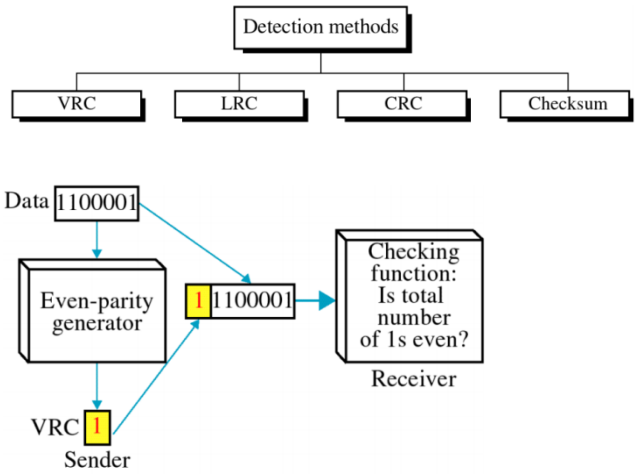
II. ERROR DETECTION

Error detection means to decide 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 for detecting errors at the destination.

Redundancy



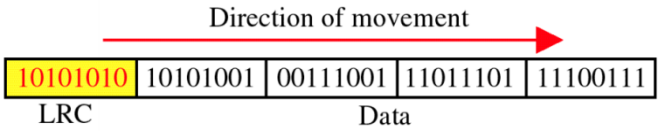
Four types of redundancy checks are used in data communication



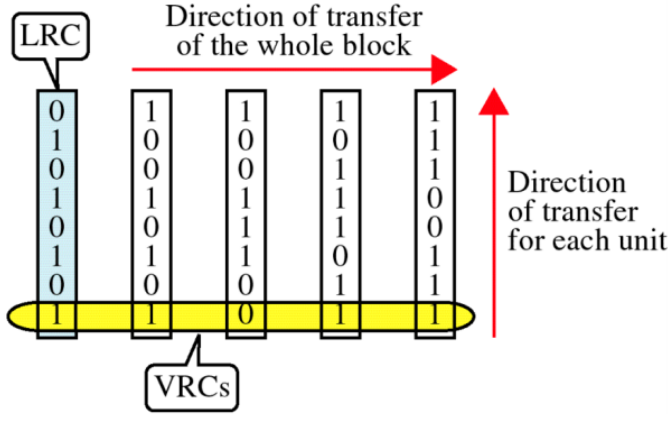
VERTICAL REDUNDENCY CHECK VRC

Performance: It can detect single bit error It can detect burst errors only if the total number of errors is odd.

Longitudinal redundancy check LRC

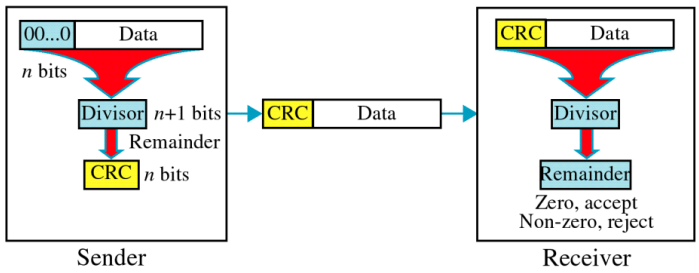


Performance: LCR increases the likelihood of detecting burst errors. If two bits in one data units are damaged and two bits in exactly the same positions in another data unit are also damaged, the LRC checker will not detect an error

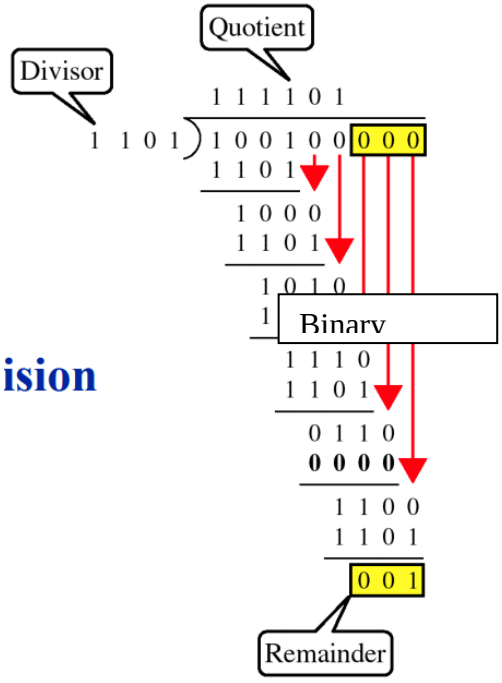


VRC & LRC

Cyclic Redundancy check CRC

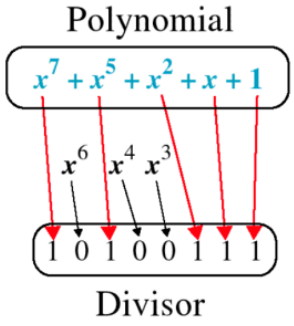


Cyclic Redundancy Check: Given a k-bit frame or message, the transmitter generates an n-bit sequence, known as a frame check sequence (FCS), so that the resulting frame, consisting of (k+n) bits, is exactly divisible by some predetermined number. • The receiver then divides the incoming frame by the same number and, if there is no remainder, assumes that there was no error



III. POLYNOMIALS

$X^7+X^5+X^3+X+1$



Standard polynomials

CRC-12

$x^{12}+x^{11}+x^3+x+1$

CRC-16

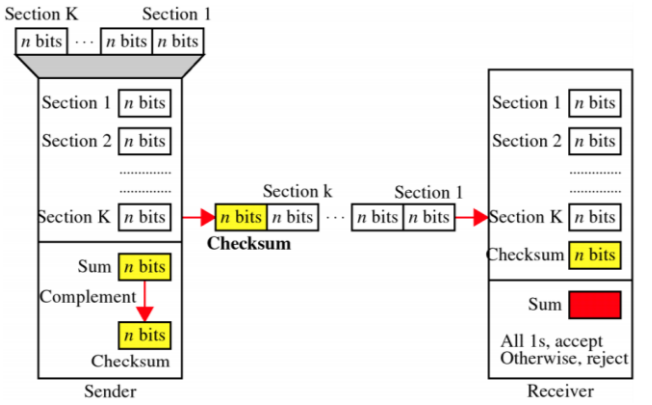
$x^{16}+x^{15}+x^2+1$

CRC-ITU

$x^{16}+x^{12}+x^5+1$

CRC-32

$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$



CHECKSUM

At the sender- Unit is divided into k sections, each of n bits. All sections are added together using one's complement to get the sum. The sum is complemented and becomes the checksum. The checksum is sent with the data.

At the receiver - unit is divided into k sections, each of n bits. All sections are added together using one's complement to get the sum. The sum is complemented. If the result is zero, the data are accepted: otherwise, they are rejected.

Performance: The checksum detects all errors involving an odd number of bits. It detects most errors involving an even number of bits. If one or more bits of a segment are damaged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect a problem.

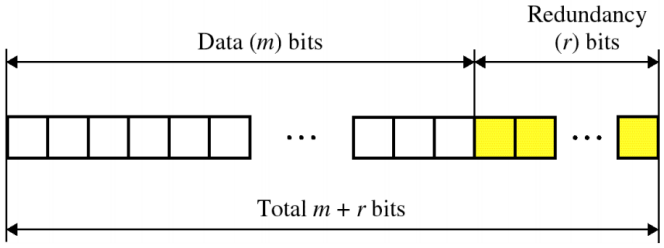
Error Correction: It can be handled in two ways:

1) Receiver can have the sender retransmit the entire data unit.

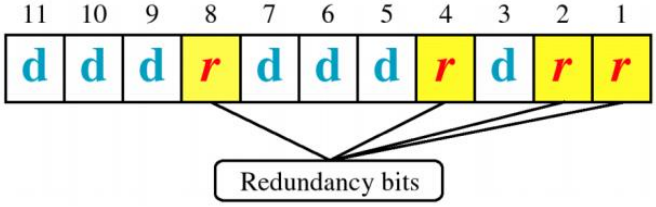
2) The receiver can use an error-correcting code, which automatically corrects certain errors.

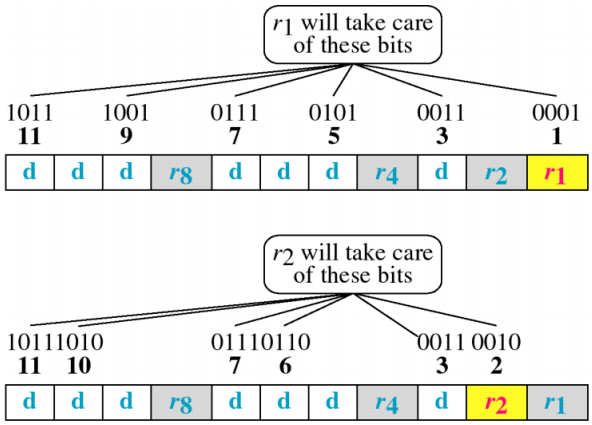
Single-bit error correction bit error correction to correct an error, the receiver reverses the value of the altered bit. To do so, it must know which bit is in error. Number of redundancy bits needed • Let data bits = m • Redundancy bits = r ∴Total message sent = m+r. The value of r must satisfy the following relation: $2^r \geq m+r+1$.

Error Correction

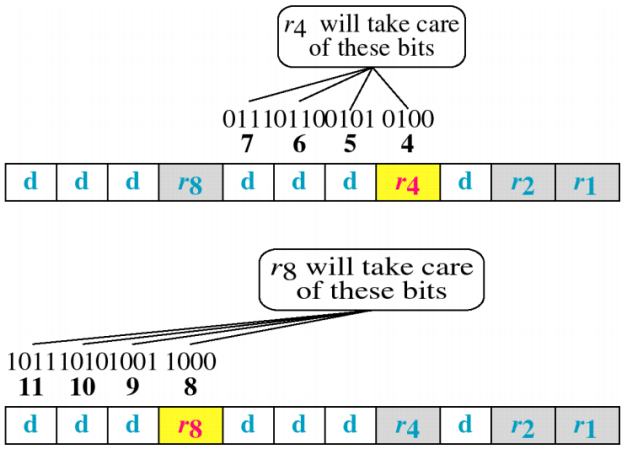


Hamming Code

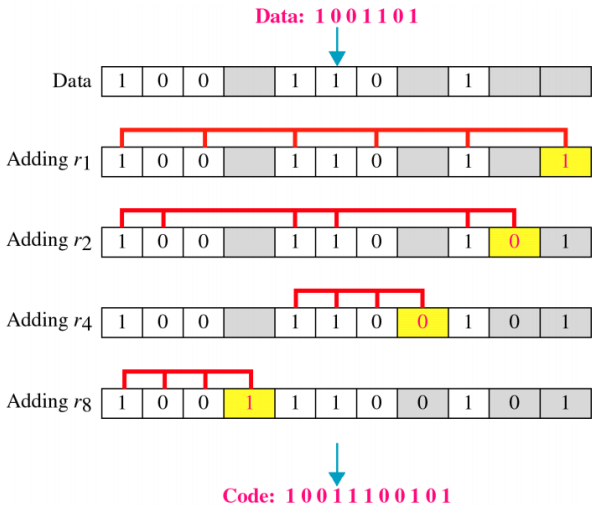




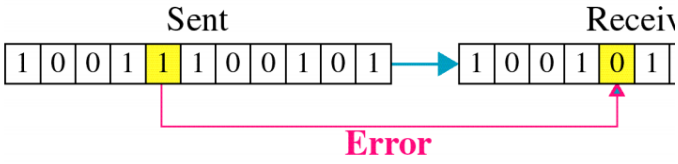
HAMMING CODE



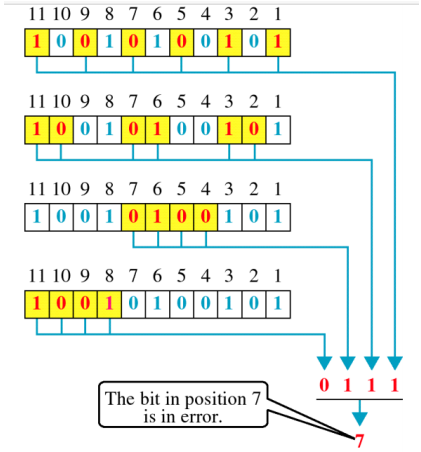
HAMMING CODE



EXAMPLE OF HAMMING CODE



SINGLE-BIT ERROR



Error Detection

Error Detecting Codes: Error detection Error detection means to decide 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 for detecting errors at the destination

Parity Checking: Parity bit means nothing but an additional bit added to the data at the transmitter before transmitting the data. Before adding the parity bit, number of 1's or zeros is calculated in the data. Based on this calculation of data an extra bit is added to the actual information / data. The addition of parity bit to the data will result in the change of data string size. This means if we have an 8 bit data, then after adding a parity bit to the data binary string it will become a 9 bit binary data string. Parity check is also called as "Vertical Redundancy Check (VRC)"

There is two types of parity bits in error detection, they are **(a)** Even parity **(b)** Odd parity

Even Parity: If the data has even number of 1's, the parity bit is 0. Ex: data is 10000001 -¬> parity bit 0, Odd number of 1's, the parity bit is 1. Ex: data is 10010001 -¬> parity bit 1,

Odd Parity: If the data has odd number of 1's, the parity bit is 0. Ex: data is 10011101 -¬> parity bit 0. Even number of 1's, the parity bit is 1. Ex: data is 10010101 -¬> parity bit 1

NOTE: The counting of data bits will include the parity bit also and the parity bits received at receiver are not equal then an error is detected. The circuit which checks the parity at receiver is called "Parity checker".

3 bit data			Message with even parity		Message with odd parity	
A	B	C	Message	Parity	Message	Parity
0	0	0	000	0	000	1
0	0	1	001	1	001	0
0	1	0	010	1	010	0
0	1	1	011	0	011	1
1	0	0	100	1	100	0
1	0	1	101	0	101	1
1	1	0	110	0	110	1
1	1	1	111	1	111	0

Message with even and odd parity

Error Detecting Techniques: This approach implemented either at Data link layer or Transport Layer of OSI Model. Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message. Basic approach used for error detection is the use of

redundancy bits, where additional bits are added to facilitate detection of errors. Some popular techniques for error detections are:- **i**. Simple Parity check **ii**. Two-dimensional Parity checks **iii**. Checksum.

IV. CONCLUSION

This paper presents errors occurs along with error-detecting code, we can also pass some data to figure out the original message from the corrupt message that we received. This type of code is called an error-correcting code. Error-correcting codes also deploy the same strategy as error-detecting codes but additionally, such codes also detect the exact location of the corrupt bit. In error-correcting codes, parity check has a simple way to detect errors along with a sophisticated mechanism to determine the corrupt bit location. Once the corrupt bit is located, its value is reverted (from 0 to 1 or 1 to 0) to get the original message.

V. REFERENCES

[1]. Thompson, Thomas M. (1983), From Error-Correcting Codes through Sphere Packings to Simple Groups, The• Carus Mathematical Monographs (#21), The Mathematical Association of America, p. vii, ISBN 0-88385-023-0

[2]. Reis A., Chang J., Vachharajani N., Rangan R., and August I., "Software Implemented Fault Tolerance," in Proceedings of the International Symposium on Code Generation and Optimization, Washington, USA, pp. 243-254, 2005.

[3]. Shannon, C.E. (1948), "A Mathematical Theory of Communication", Bell System Technical Journal, p. 418, 27•

[4]. Golay, Marcel J. E. (1949), "Notes on Digital Coding", Proc.I.R.E. (I.E.E.E.), p. 657, 37•

[5]. Nicolescu B., Savaria Y., and Velazco R., "Sied: Software Implemented Error Detection," in Proceedings of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Boston, USA, pp. 589-596, 2003

[6]. Gupta, Vikas; Verma, Chanderkant (November 2012). "Error Detection and Correction: An Introduction" (PDF). International Journal of Advanced Research in Computer Science and Software Engineering. 2 (11). Retrieved August 21, 2019.

[7]. J. Mc Auley, Reliable Broadband Communication Using a Burst Erasure Correcting Code, ACM SIGCOMM, 1990. Frank van Gerwen. "Numbers (and other mysterious) stations". Retrieved 12 March 2012.•

[8]. Asghari A., Taheri H., Pedram H., and Kaynak O., "Software-based Control Flow Checking Against Transient Faults in Industrial Environments," IEEE Transactions on Industrial Informatics, vol. no. 99, pp. 1, 2013.

[9]. Gary Cutlack (25 August 2010). "Mysterious Russian 'Numbers Station' Changes Broadcast After 20 Years". Gizmodo. Retrieved 12 March 2012.

[10]. Ben-Gal I.; Herer Y.; Raz T. (2003). "Self-correcting inspection procedure under inspection errors" (PDF). IIE Transactions on Quality and Reliability, 34(6), pp. 529-540.

[11]. K. Andrews et al., The Development of Turbo and LDPC Codes for Deep-Space Applications, Proceedings of the IEEE, Vol. 95, No. 11, Nov. 2007.

[12]. Li A. and Hong B., "On-line Control Flow Error Detection using Relationship Signatures Among Basic Blocks," Elsevier journal of Computers and Electrical Engineering, vol. 36, no.1, pp. 132-141, 2010.

[13]. Li A. and Hong B., "Software Implemented Transient Fault Detection in Space Computer," Elsevier journal of Aerospace Science and Technology, vol. 11, no. 2, pp. 245-252, 2007.

[14]. Huffman, William Cary; Pless, Vera S. (2003). Fundamentals of Error-Correcting Codes. Cambridge University Press. ISBN 978-0-521-78280-7.
[15]. Scott A. Moulton "A Survey of Techniques for Improving Error-Resilience of DRAM", Journal of systems architecture, 2018
[16]. "Using StrongArm SA-1110 in the On-Board Computer of Nanosatellite". Tsinghua Space Center, Tsinghua University, Beijing. Retrieved 2009-02-16.[permanent dead link]
[17]. Jeff Layton. "Error Detection and Correction". Linux Magazine. Retrieved 2014-08-12.·
[18]. "Documentation/edac.txt". Linux kernel documentation. kernel.org. 2014-06-16. Archived from the original on 2009-09-05. Retrieved 2014-08-12.
[19]. Behrouz A. Forouzan, “Data Communication and Networking”, 3 rd Edition, Tata McGraw Hill, 2004.·
[20]. William Stallings, “Data and Computer Communications”, 8th Edition, Pearson Education, 2007.·
[21]. Mammeri S. and Beghdad A., “On Handling Real-time Communications in MAC Protocol,” International Arab Journal of Infoormation Technology, vol. 9, no. 5, pp. 428-435, 2012.