International Association of Scientific Innovation and Research (IASIR)
(An Association Unifying the Sciences, Engineering, and Applied Research)

ISSN (Print): 2279-0047
ISSN (Online): 2279-0055

International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

CRC AND LOOK-UP TABLE ASSISTED ERROR CORRECTION IN A CONVOLUTIONAL CODED SYSTEM ON DSP

S. V. Viraktamath[1], Dr. Girish V. Attimarad[2]
[1] Assistant Professor, Department of E&CE, SDMCET, Dharwad, Karnataka, India.
[2] Professor, HOD Department of E&CE, Dayanand Sagar College of Engineering, Bangalore, Karnataka, India

***Abstract**: Cyclic Redundancy Codes (CRC) code provides a simple, yet powerful, method for the detection of errors during digital data transmission and storage. Among Forward Error Correction (FEC) schemes, convolutional encoding and Viterbi decoding are the most popular because of their powerful coding-gain performances. In this paper the implementation of CRC and Viterbi decoder on DSP is presented. CRC-32 and Viterbi hard decision decoding algorithm for rate 1/2 and for different generator polynomials is simulated and also implemented on DSP-TMS320C5416. Also the for higher SNR receivers concept of serially concatenated CRC- Convolutional Coding (CC) with lookup table at the decoder side is proposed.*

***Key Words**: CRC, DSP, Viterbi, Trellis, Constraint length.*

I. INTRODUCTION

The evolving world of telecommunications requires increasing reliability and speed in communications. Reliability in information storage and transmission is provided by coding techniques. Information is usually coded in bit streams and transmitted over the communication medium, channel. The communication media is prone to errors due to noise present in the analog portion of the channel. Therefore errors have to be detected and corrected while decoding. CRC has the advantages of easy coding and decoding as well as strong abilities of checking errors and correcting errors. Therefore, it was widely used in the field of communications. Reliability in information storage and transmission is provided by coding techniques. CRC is an error-detecting code designed to detect accidental changes to raw computer data, and is commonly used in digital networks and storage devices such as hard disk drives. Blocks of data entering these systems get a short check value attached, derived from the remainder of a polynomial division of their contents; on retrieval the calculation is repeated, and corrective action can be taken against presumed data corruption if the check values do not match. The CRC was invented by W. Wesley Peterson in 1961. CRC is an error detecting code that is widely used to detect corruption in blocks of data that have been transmitted or stored. The Error Control Coding techniques (ECC) rely on the systematic addition of redundant bits at the transmitting side. The task of channel coding is to encode information sent over a communication channel in such a way that in the presence of channel noise, errors can be detected and possibly corrected. There are two coding methods - backward error correction codes and forward error correction codes. Backward error correction codes requires only error detection, if an error is found then the transmitter is requested to retransmit the message. Forward error correction codes require the decoder to be capable of correcting errors. There are several error correcting codes and these are classified under two basic categories namely block codes and convolutional codes.

Convolutional codes [1] differ from block codes [2] in the sense that bit streams are not partitioned into binary words instead redundancy is added continuously to the whole stream. Convolutional codes are widely used error control coding technique in channel coding because of low complexity and error controlling capability. Viterbi decoding algorithm [3, 4] is the simplest and best algorithm for decoding of convolutional codes. The Viterbi algorithm first appeared in the coding literature in a paper written by Andrew J. Viterbi in 1967 [5]. Since then, due to its easiness in implementation, it has been applied to many different areas related to decoding problems. The 8-bit parallel CRC-32 is proposed in [6] to meet the high throughput of USB3.0. Exhaustive survey of all CRC polynomials from 3 bits to 15 bits is presented in [7]. A set of 35 new polynomials in addition to 13 previously published polynomials are also described. The method that realizes the ability of multiple bits error correction using cyclic redundancy check codes is presented in [8]. The structures of 8-bit CRC are presented in [9]. The joint decoding scheme of serially concatenated CRC and convolutional code (CC) has been investigated in [10]. This paper is organized as follows. Section 2 gives the proposed work. Section 3 gives the CRC coding. The Viterbi decoder is discussed in section 4. Processor TMS 320VC5416 is discussed in section 5. The results of the proposed model are discussed in section 6. Next section concludes the paper.

II. PROPOSED WORK

The basic block diagram of a system used for the simulation is shown in Fig.1. The output of the source encoder is given to the convolutional encoder. The output of the convolutional encoder is given to the CRC encoding;

the output of which will be modulated and sent on the channel. At the receiver the reverse process is done. The CRC encoding, decoding, convolution encoding and decoding blocks are simulated and implemented on DSP.

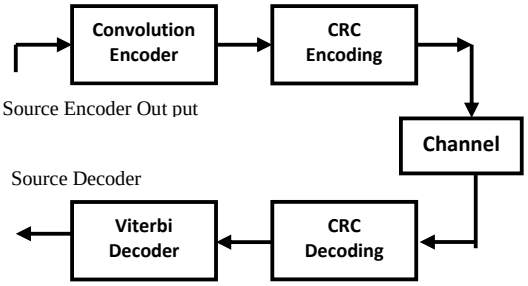


Fig.1 Block diagram of an proposed system

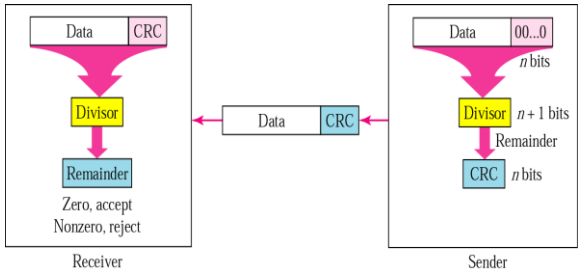


Fig. 2 Block diagram of CRC

III. CRC CODING

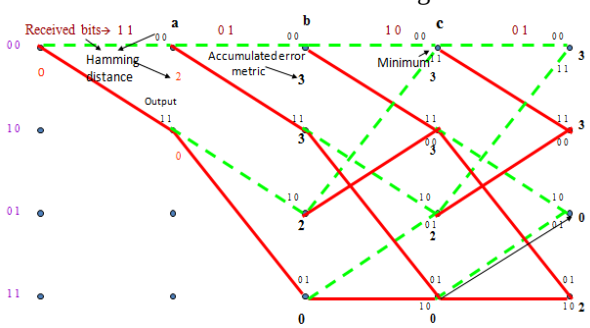
CRC are specifically designed to protect against common types of errors on communication channels, where they can provide quick and reasonable assurance of the integrity of messages delivered as shown in Fig.2. However, they are not suitable for protecting against intentional alteration of data. The selection of generator polynomial is the most important part of implementing the CRC algorithm. The polynomial must be chosen to maximize the error-detecting capabilities while minimizing overall collision probabilities. CRC is divided into the following types: Code CRC-12, code CRC-16, code CRC-CCITT, and code CRC-32. Code CRC-12 is usually used to send 6-bit string. Code CRC-16 and code CRCCCITT is used to send 8-bit string, and code CRC-16 is mainly used in America, however, code CRC-CCITT is often used in European countries.

Code CRC-32 is often used in a kind of synchronous transfer which is called Point to-Point transfer. In the proposed system CRC-32 is simulated and implemented. The Convolutional encoder described in this system is of rate 1/2, and of constraint length -3. Programming is done in 'C' language for both coding and decoding and the same is implemented on DSP-5416 kit. Input to encoder is entered by the user. The output of the encoder is stored in the matrix form, and the same is fed to the decoder part.

IV. VITERBI DECODER

For the implementation of Viterbi decoder algorithm, there are two techniques available: the soft decision decoding and hard decision decoding method. In case of hard decision, the decoder makes a firm or hard decision as to whether one or zero is transmitted and provides no other information regarding how reliable the decision is, hence, its output is only either zero or one (the output is quantized only to two levels) which are called hard-bits. The soft decoder provides the system with some side information together with the decision. The side information provides the decoder with a measure of confidence for the decision. The decoder outputs, which are called soft-bits, are quantized to more than two levels. In this paper the simulation and implementation of Viterbi decoder algorithm is discussed for hard decision decoding method.

The input to and output from the system are bit streams. The bits entered by the user are used as message/input bit stream to convolutional encoder. A (n, k, m) convolutional encoder accepts k-bit blocks of input sequence and produces n-bit blocks of output sequence. It consists of 'm' k-stage shift registers and 'n' modulo-2 adders. The outputs of 'n' modulo-2 adders are sequentially sampled to produce the encoded sequence. A (2, 1, 2) convolutional encoder is considered for simulation and implementation. While decoding the proposed code generates output table and next state table depending on generator polynomial and stores them. Basically, decoding of convolutional codes is comparison of different paths in trellis. The trellis diagram for (2, 1, 2) convolutional encoder is as shown in Fig.3.



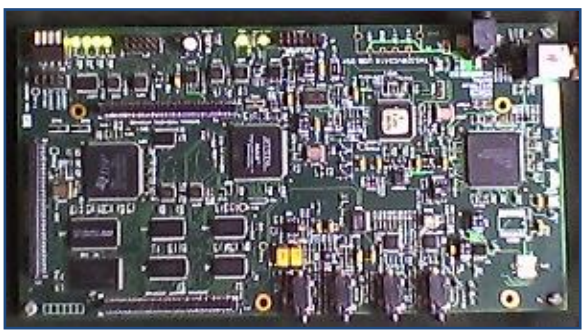


Fig.3. Trellis for convolutional encoder (2, 1, 2). Fig. 4 The TMS320VC5416 kit

In hard decision decoding technique, the Hamming distance is computed by simply counting how many bits are different between the received encoded bits and the actual output bits. The Hamming distance values are computed at each time instant for the paths between the states at the previous time instant and the states at the current time instant are called 'branch metrics'. For the first time instant these results are saved as "Accumulated Error Metric" values, associated with the states. For the second time instant on, the accumulated error metrics will be computed by adding the previous accumulated error metrics to the current branch metrics. When two paths enter the same state, the one having the best metric (i.e. lower branch metric) is chosen, this path is called the 'surviving path'. Then by seeing the path and using output table and next state table decoding is done. This procedure is repeated for all encoded bits and at every comparison makes a hard decision as to whether one or zero is transmitted. The output of hard decision decoder is compared with original message bits for verification.

V. PROCESSOR TMS 320VC5416

The TMS320VC5416 fixed-point, digital signal processor (DSP) is based on an advanced modified Harvard architecture that has one program memory bus and three data memory buses. This processor provides an arithmetic logic unit (ALU) with a high degree of parallelism, application-specific hardware logic, on-chip memory, and additional on-chip peripherals. The basis of the operational flexibility and speed of this DSP is a highly specialized instruction set. Separate program and data spaces allow simultaneous access to program instructions and data, providing a high degree of parallelism. Two read operations and one write operation can be performed in a single cycle. Instructions with parallel store and application-specific instructions can fully utilize this architecture. In addition, data can be transferred between data and program spaces. Such parallelism supports a powerful set of arithmetic, logic, and bit-manipulation operations that can all be performed in a single machine cycle. The device also includes the control mechanisms to manage interrupts, repeated operations, and function calls. The TMS320VC5416 DSP starter kit (DSK) is a low-cost development platform designed to speed the development of power-efficient applications based on TI's TMS320VC54x DSPs. The kit, which provides new performance-enhancing features such as USB communications and true plug-and-play functionality, gives both experienced and novice designers an easy way to get started immediately with innovative product designs. The C5416 DSK offers the ability to detect, diagnose and correct DSK communications issues, download and step through code faster and get a higher throughput with Real Time Data Exchange (RTDX™). The contents of the kit include: C5416 DSK Code Composer Studio™ v2.1 IDE, Quick Start Guide, Technical Reference, Customer Support Guide, USB Cable, Universal Power Supply and AC Power Cord(s). The Fig.4 shows the view of TMS320VC5416 kit.

VI. RESULTS AND DISCUSSIONS

In this section, simulation results are presented. Different types of CRC are simulated from CRC-8 to CRC-32. The simulation results of CRC-8 and CRC 32 are shown in the paper. Fig 5 shows the simulation environment completely.

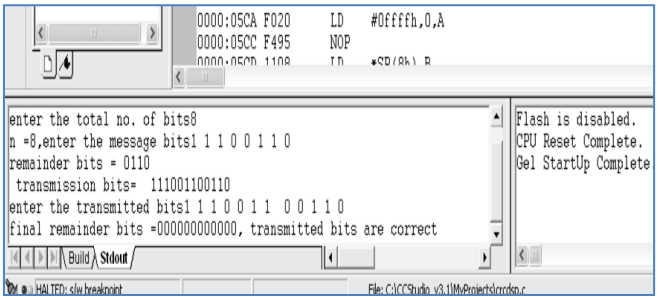


Fig. 5 Simulation Environment of DSP kit

The bottom left window called 'stdout' shows the input and output of the program. The simulation result of CRC 8 is shown in Fig 6 when received message has an error. Fig 7 shows the output when the received message has an error. Fig 8 and Fig 9 shows the input and output of the CRC -32 with errors as well as without errors.

```
the message bits1110011001000111
 check bits = 11111101
 transmission bits=  11100110010
the received bits are=1110011001
final remainder bits =0000000000
transmitted message has error
```

Fig. 6 CRC-8 with error in the received message

```
the message bits111001100100011
 check bits = 11111101
 transmission bits=  1110011001
the received bits are=111001100
final remainder bits =000000000
 transmitted bits are correct
```

Fig. 7 CRC-8 with correct received message

```
the message bits1110011001000
 check bits = 0011011111000001
 transmission bits=  11100110
the received bits are=1110011
final remainder bits =0000000
transmitted message has error
```

Fig. 8 CRC-32 with error in the received message

```
the message bits1110011001000
 check bits = 0011011111000001
 transmission bits=  11100110
the received bits are=1110011
final remainder bits =0000000
 transmitted bits are correct
```

Fig. 9 CRC-32 with correct received message

The code has been verified for all possible lengths of message polynomial and generator polynomial. The CRC-8 to CRC-32 are successfully implemented on TMS320VC5416 fixed-point DSP.

The simulation and implementation of convolutional encoder and Viterbi decoder is carried out for different message bits and different generator polynomials. The processor implementation results for rate ½ convolutional encoder and Viterbi decoder with generator polynomials g1= {1 1 1} and g2= {1 0 1} and for message bits (1 1 1 0) are as shown in Fig.10 and Fig.11 respectively.

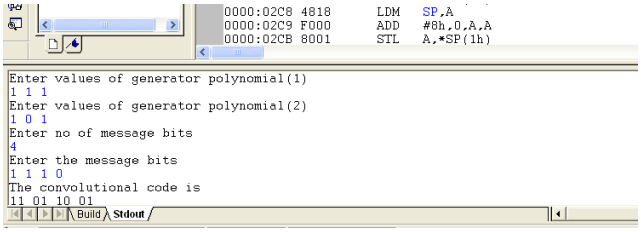


Fig.10. Implementation result of convolutional encoder for message stream (1 1 1 0)

Fig.11. Implementation result of Viterbi decoder with one bit error for message stream (1 1 1 0)

Similarly the results are verified for generator polynomials by giving different message bits as input. The simulation results of Viterbi decoder for generator polynomials g1= {1 1 1}, g2= { 1 1 0} and g1={1 1 0}, g2={1 0 1} are as shown in Fig.12 and Fig.13 respectively.

Fig.12.Implementation result of Viterbi decoder with two bits error for message stream (1 0 0 1 1 0) for Generator polynomial g1={1 1 1} and g2={1 1 0}

Fig.13. Implementation result of Viterbi decoder with two bits error for message stream (1 1 0 1 0 1) for Generator polynomial g1={1 1 0} and g2={1 0 1}

Also different generator polynomials are given as input to the encoder block. The simulation and implementation results exactly matched with the theoretical results. The proposed algorithm is working for all message bits and delivering the expected results. In this paper separate results for CRC and Viterbi decoder are shown. Many researchers suggested ideas how to reduce the computations by selecting only few branch metric [10, 11] based on the hamming distances. The investigation on joint decoding schemes for serially concatenated cyclic redundancy check (CRC) and convolutional code (CC) is done in [12]. A soft decoding algorithm for CC decoding is used with CRC to check the hard decision output from the soft CC decoding module for errors. After Viterbi decoding to check the errors CRC was used. The use of CRC was after decoding [12]. It is possible to use CRC before the Viterbi decoding for the errors; if no errors found just using the lookup table decoding can be done without going for the online computation using trellis. The lookup table will be having the decoded output offline. This saves the time as well as computations. If the errors were found then classical decoding of Viterbi decoder may be used.

VII. CONCLUSIONS

The procedure of calculation of remainder or redundant bits at the transmitter and checking the errors at the receiver is presented in the paper. The CRC code has been verified for all possible lengths of message polynomial and generator polynomial. The CRC-8 to CRC-32 are successfully implemented on TMS320VC5416 fixed-point DSP. The simulation results are presented in the paper. Convolutional Encoder and Viterbi decoder for the rate ½ is simulated for different constraint lengths. From the simulation it is found that the error recovery capability of the Viterbi decoder varies from generator polynomials as well as for different constraint lengths. Convolutional encoder and Viterbi decoder are successfully implemented on DSP-TMS320C5416. The usage of look-up table may help in fast decoding.

ACKNOWLEDGEMENTS

The authors thank the authorities of Sri Dharmasthala Manjunatheshwara College of Engineering and Technology, Dhavalagiri, Dharwad, and authorities of TEQIP 1.2 for encouraging us for this research work and providing financial support.

REFERENCES

[1] J. Viterbi, "Convolutional codes and their performance in communication systems", IEEE Transaction Communication Technology,cVol.19, pp. 751-772, Oct. 1971.
[2] B. Sklar, Digital Communications: "Fundamentals and Applications", 2nd edition, Prentice-Hall, Upper Saddle River, N J, 2001
[3] G. C. Clark Jr. and J. B. Cain, "Error-Correction Coding for Digital Communications", Plenum Press, NY, 1981.
[4] A. J. Viterbi and J. K. Omura, "Principles of Digital Communications and Coding", McGraw-Hill, NY, 1979.
[5] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", IEEE. Transaction of Information Theory, vol. IT-13, pp. 260-269, Apr. 1967.
[6] Julian Satran, Dafna Sheinwald and Ilan Shimony "Brief Contributions- Out of Order Incremental CRC Computation", IEEE Transactions on computers, VOL. 54, NO. 9, SEPTEMBER 2005
[7] Ma Youjie, Zhang Haitao, Zhou Xuesong, Qi Ming, Xu Lijin, "The Realization of the CRC Arithmetic which is based on DSP", 2009
[8] Philip Koopman, Tridib Chakravarty, "Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks", The International Conference on Dependable Systems and Networks, DSN-2004.
[9] Sunil Shukla, Neil W. Bergmann, "Single bit error correction implementation in CRC-16 on FPGA", ICFPT 2004, 0-7803-8652-3/04/$20.00 0 2004 IEEE.
[10] K. S. Arunlal1 and Dr. S. A. Hariprasad2 "An Efficient Viterbi Decoder", International Journal of Advanced Information Technology (IJAIT) Vol. 2, No.1, February 2012. DOI : 10.5121/ijcsea.2012.2109 95
[11] Renqiu Wang, Wanlun Zhao , and Georgios B. Giannakis, "*CRC-Assisted Error Correction in a Convolutionally Coded System", IEEE Transactions On Communications. Vol 56, No 11, November 2008. Pages -1807-1815.*

BIOGRAPHIES

Mr. S. V. Viraktamath is with SDMCET, Dharwa. He has received a gold medal from VTU Belgaum for securing first rank in M.Tech (DC&N). He is the Life Member of IETE, IE and ISTE. He has served as a reviewer for many International conferences and journals. His research interests include Error control coding, Wireless communication and Networking.

Dr. G. V. Attimarad is with Dayanand Sagar College of Engineering Bangalore. His research interest includes area of waveguides and wireless communications. He has published many papers in reputed National and International journals. He is the Life Member of IETE, IE and ISTE. He has served as a reviewer for many International conferences.