Viterbi Decoding in Field Programmable Gate Arrays for the Error Correction in Data Communication

Nitin S. Sonar[1] and Faris S. Al-Naimy[2]

[1]*Faculty, Ibra College of Technology,*
Directorate of Technological Education,
Ministry of Manpower, Sultanate of Oman
E-mail: ns.sonar@gmail.com
[2]*HOD Engineering, Ibra College of Technology,*
Directorate of Technological Education,
Ministry of Manpower, Sultanate of Oman
E-mail: hodengg@ict.edu.om

Abstract

Forward error correction ((FEC) plays an important role in the field of telecommunication and information theory as it improve the capacity of a channel. It has been observed that Convolutional encoding with Viterbi decoding is a powerful method for error detection and correction. It is suited to a channel in which the transmitted signal is corrupted mainly by additive white Gaussian noise (AWGN). In this paper, the design and implementation of a novel reconfigurable Viterbi decoder is presented.

The synthesized has been done on Xilinx SPARTAN – 3E FPGA for proposed decoder to support code rates 1/2 and 1/3 with a throughput of 20 Mbps.

Index Terms: Forward error correction; Viterbi decoding; Field Programmable Gate Arrays.

Introduction

In information theory and coding theory with applications in computer science and telecommunication, error detection and correction or error control are techniques that

enable reliable delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. Error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data. The use of error-correcting codes has proven to be an effective way to overcome data corruption in digital wireless communication channels, enabling reliable transmission over noisy and fading channel. This requires low power decoders as they consume lot of power. Power reduction in any system can be achieved at device level, at circuit level or at architectural level. In this paper, power reduction is achieved at architecture level.[1]

The channel code is used to protect data sent over it for storage or retrieval even in the presence of noise (errors). Channel coding is the process of adding redundant information to the data being transmitted. The reason why we are using the channel coding is because the transmitted signal are often corrupted by channel noise and we can use channel coding to correct these errors so as to achieve the reliable transmission of information over a large verities of channels. The advantages of forward error correction are that a back-channel is not required and retransmission of data can often be avoided (at the cost of higher bandwidth requirements, on average). FEC is therefore applied in situations where retransmissions are relatively costly or impossible. In particular, FEC information is usually added to most mass storage devices to protect against damage to the stored data. Viterbi decoding is a technique for performing maximum likelihood sequence detection on data that has been convolutionally coded. The decoding problem is to determine through the trellis, with path matric being defined as the sum of the branch metrics along the path. This is done in a stepwise manner by processing a set of states metrics forward in time, stage by stage over the trellis.

Architecture of the error corrector

Architecture of the error corrector transreceiver is as shown in the Figure 1. In the transmitter section initially clock is divided to a suitable clock. The start signal for encoding is given after the delay of 16 mSec for key de bounce. Thereafter data is encoded using convolutional encoder. Then the encoded data will be transmitted. The acceptor block of the receiver will detect the start bit and then it will send the data to the Viterbi decoder. The decoder will give out the original information transmitted. A pre-computation architecture incorporated with Trellis-algorithm for Viterbi Decoder can effectively reduce the power consumption without degrading the decoding speed much.[2]
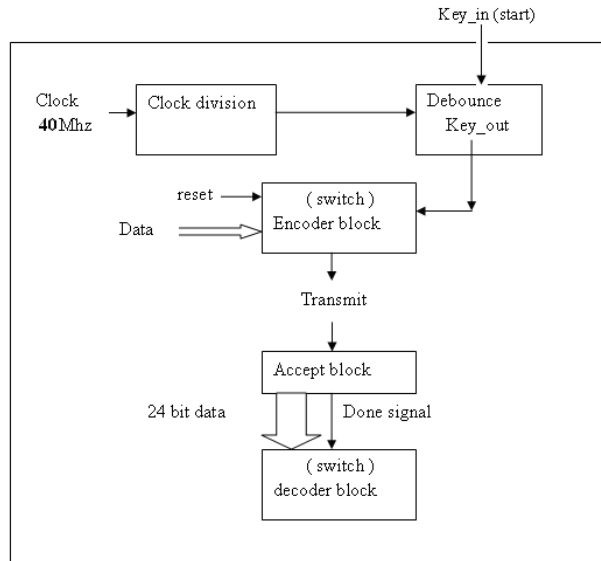
Figure 1: Block Diagram showing architecture of the error corrector.

Transmitter Section

The transmitter clock division block will Generate 20 MHz clock from the clock of the FPGA kit. After giving 16msec delay for key de-bounce, the start signal will be given for encoding. We can take the input data in the form of 8 bits and then the Convolution encoder will encode the data. Turbo codes have been shown to yield very good performance for long block sizes. However, their performance degrades when the frame size decreases.[3] The encoded data will be transmitted serially in asynchronous mode of transmission. Start and stop bits are also added while transmission.
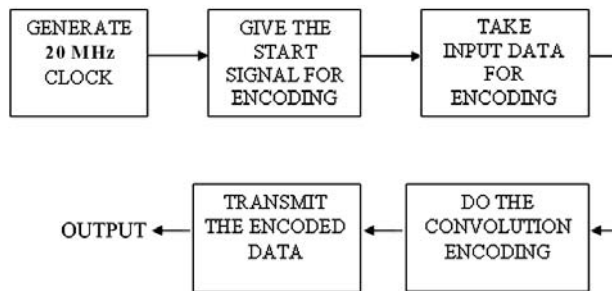

Figure 2: Block Diagram of the Transmitter

Receiver Section

The receiver section has four blocks as demonstrated in Figure 3. The receiver will wait for falling edge to be detected, after detecting falling edge it will take sample

with a delay of 52μsec. If sampled bit is 0 then start bit is being detected. Then it will sample the next 25 bits after every 104 μsec. When it will receive 25 bits it will check that next bit received is STOP bit. The received data will be given to the Viterbi decoder for decoding. The decoder will give out the original information transmitted.
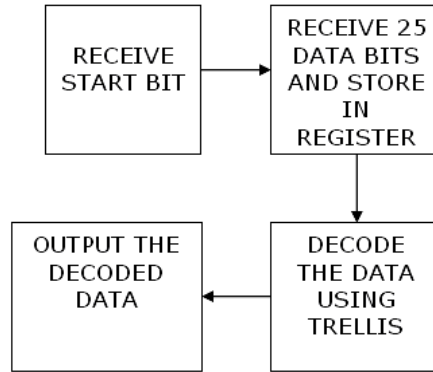
Figure 3: Block Diagram of the Receiver

Convolution Encoder

A convolutional coding is done by combining the fixed number of inputs bits. The input bits are stored in the fixed length shift register and they are combined with the help of mod-2 adders. This operation is equivalent to binary convolution and hence it is called convolution coding. Convolutional coding and Viterbi decoding, along with binary phase-shift keyed modulation, is presented as an efficient system for reliable communication on power limited satellite and space channels. Performance results, obtained theoretically and through computer simulation, are given for optimum short constraint length codes for a range of code constraint lengths and code rates.[4] The concept is illustrated with help of above example.

As shown in Figure. 4, whenever the message bit is shifted to position 'S', the new values of V1 and V2 are generated depending upon S0, S1, and S2. S1 and S2 store the previous two message bits. The current bit is present in S0. Thus, equation becomes,

$$V1 = S0 \text{ XOR } S2, \text{ and}$$
$$V2 = (S0 \text{ XOR } S1) \text{ XOR } S2$$

The output switch first samples V1 and then V2. The shift register then shifts contents of S1 to S2 and contents of S0 to S1. Next input bit is then taken and stored in S0. Again V1 and V2 are generated according to this new combination of S0, S1, and S2. The output switch then samples V1 and V2.
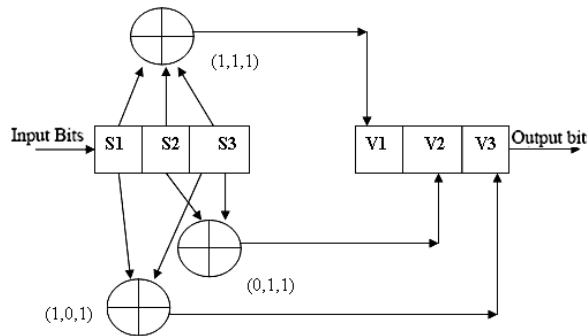
For every input message bit two encoded output bits V1 and V2 are transmitted. For a single message bit, the encoded code word is two bits for this convolution encoder.

Number of message bits k=1,
Number of encoded output bits for one message bit, n=2.
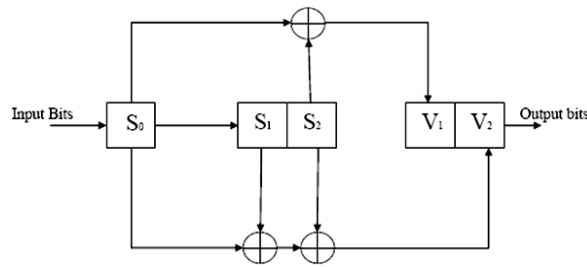
Figure 4: The rate 1/2 Convolutional encoder
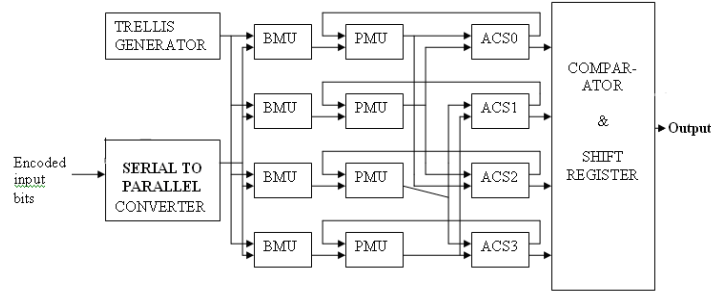

Figure 5: The rate 1/3 Convolutional encoder

Viterbi Algorithm

The Viterbi algorithm (VA) is known as a maximum likelihood (ML)-decoding algorithm for convolutional codes. Maximum likelihood decoding means finding the code branch in the code trellis that was most likely to be transmitted. Therefore, maximum likelihood decoding is based on calculating the hamming distance for each branch forming encode word. The most likely path through the trellis will maximize this metric. The trace back and decoder can simultaneously work in order to improve the processing speed.[5]

Viterbi algorithm performs ML decoding by reducing its complexity. It eliminates least likely trellis path at each transmission stage and reduce decoding complexity with early rejection of unlike paths. Viterbi algorithm gets its efficiency via concentrating on survival paths of the trellis. The Viterbi algorithm is an optimum algorithm for estimating the state sequence of a finite state process, given a set of noisy observations.

The implementation of the VA consists of three parts: branch metric computation, path metric computations unit computes a number of recursive equations. In a Viterbi decoder (VD) for an N-state convolutional code, N recursive equations are computed at each time step. Existing high-speed architectures use one processor per recursion equation. The main drawback of these Viterbi Decoders is that they are very expensive in terms of chip area. In current implementations, at least a single chip is dedicated to the hardware realization of the Viterbi decoding algorithm.

Viterbi Decoder

Figure 6: Decoder block schematic

The most commonly used decoding algorithm for convolutional codes is the Viterbi algorithm, which is a Maximum likelihood sequence estimator. The Viterbi algorithm is essentially a shortest path algorithm for computing the shortest path through the trellis associated with code the receiver receives the serial encoded data and converts it into 2-bit parallel data. This parallel data is now feed to the subsequent blocks. Trellis Generator generates the predefined sequence which is used to calculate the branch metric unit. The effects of quantization and path metric memory in practical Viterbi decoding implementations shows that in systems with limited antenna diversity, low-memory codes achieve a better error-rate performance compared to that of high-memory codes.[6]

Serial to parallel converter: The input bits to the decoder are serial in nature. The output of encoder is n bits for one input bit. As the input bits to the decoder are serial in nature, therefore to get required n bits we convert the serially obtained bits into n bit parallel form. For Encoder the input bits are converted to 2 bit parallel form. The output of serial to parallel converter is to be compared with all the possible output of the code trellis.

Branch metric unit: The main function of this unit is the calculation of the branch metric for each input to the decoder. As trellis diagram is fixed for the particular decoder, from this trellis logical equation are derived to calculate branch metric
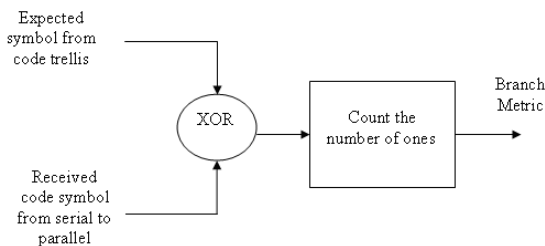
Figure 7: Branch metric unit

Path metric unit: This unit has two inputs one from the Branch metric unit and second is the previous path metric from ACS unit. This unit just adds the two inputs and creates the new path metric.

ACS (Add Compare & Store unit): For this unit there are two input path metrics. ACS compares the two path metrics and gives out the minimum path metric at the output with its data bit. The data bits are stored.

Comparator and shift register: This unit accepts the path metric selected by every ACS unit for every cycle and gives out minimum path metric with its data bits.

Conclusion

We have proposed a new Viterbi decoder architecture which allows for run-time adaptation to different code rates. The new architecture is suitable for high data rate decoding. The architecture was implemented on a Xilinx FPGA. With a throughput of 20 Mbps, the proposed decoder is suitable for use in receiver architecture of 3G cellular code division multiple access environments. We believe that such an architecture has considerable application in future networks where different decoding standards shall have to be supported on a single device.

References

[1] Joshi, S.P. Paily, Dept. of Electron. & Commun. Eng., Indian Inst. of Technol. Guwahati, Guwahati, India, This paper appears in: Communications and Signal Processing (ICCSP), 2011 International Conference, Issue Date: 10-12 Feb. 2011, On page(s): 499 – 503, Location: Kerala, Print ISBN: 978-1-4244-9798-0 INSPEC Accession Number: 11887605, Digital Object Identifier: 10.1109/ICCSP.2011.50739371, Date of Current Version: 24 March 2011.4

[2] He, J. Liu, H. Wang, Z. Huang, X. Zhang, K. School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA, This paper appears in: Very Large Scale Integration (VLSI) Systems, IEEE Transactions. Volume: PP, Issue:99, On page(s): 1, ISSN: 1063-8210, Digital Object Identifier: 10.1109/TVLSI.2011.2111392, Date of Publication: 06 March 2011.

[3] Fan Yang Zhendong Luo Baoyu Tian Yubai Li Sch. of Commun. & Inf. Eng., Univ. of Electron. Sci. & Technol. of China, Chengdu, China, This paper appears in: Communications Letters, IEEE, Issue Date: February 2010, Volume: 14, Issue:2, On page(s): 172, ISSN: 1089-7798, INSPEC Accession Number: 11135900, Digital Object Identifier: 10.1109/LCOMM.2010.02. 090989, Date of Current Version: 05 February 2010.

[4] Mark, J.; Maxemchuk, N.; Ziemer, R.; Taylor, D.; Tranter, W.; Book Title: The Best of the Best:Fifty Years of Communications and Networking Research Publisher: IEEE, Pages: 361 -374, Edition: 1, Subject Categories:

Communication, Networking & Broadcasting, Publication Year: 2010, Copyright Year: 2007, ISBN: 9780470546543.

[5] Lei-ou Wang Zhe-ying Li, Inst. of Micro-Electron. Applic. Tech, Beijing Union Univ., Beijing, China. This paper appears in: Artificial Intelligence and Education (ICAIE), 2010 International Conference. Issue Date: 29-30 Oct. 2010, On page(s): 77, Print ISBN: 978-1-4244-6935-2, INSPEC Accession Number: 11662520 Digital Object Identifier: 10.1109/ICAIE.2010.5641528, Date of Current Version: 18 November 2010.

[6] Chatzigeorgiou, I. Demosthenous, A. Rodrigues, M.R.D. Wassell, I.J, Digital Technol. Group, Univ. of Cambridge, Cambridge, UK, This paper appears in: Communications, IET, Issue Date: March 5 2010, Volume: 4, Issue:4, On page(s): 419, ISSN: 1751-8628, INSPEC Accession Number: 11243903 Digital Object Identifier: 10.1049/iet-com.2008.0596 Date of Current Version: 25 February 2010.