IJCSN International Journal of Computer Science and Network, Volume 4, Issue 6, December 2015
ISSN (Online): 2277-5420 www.IJCSN.org
Impact Factor: 0.417

Design and Implementation of Convolution Encoder and Viterbi Decoder

[1] Rakhi B. Menon, [2] Dr. Gnana Sheela K

[1, 2] Toc H Institute of Science and Technology (CUSAT)
Arakkunnam, Ernakulam-682313

Abstract - Data transmissions over wireless channels are affected by attenuation, distortion, interference and noise, which affect the receiver's ability to receive correct information. Convolution encoding with Viterbi decoding is a powerful method for forward error correction. Convolution encoders and Viterbi decoders play an important role in digital communication especially, when channel is noisy and introduces errors in transmitted signal. The use of re-transmission methods is not efficient and has large latency measure up to the rising speed and data rates of communication links, the need of new techniques arise here to be compatible with those systems. Convolution encoding with forward error correction Viterbi decoding is designed. Implementation parameters for the decoder have been determined through simulation and the decoder should be implemented on a Xilinx FPGA SPARTAN 3E Kit. Verilog HDL language is used as a design entry.

Keywords - *Convolution Encoder, Viterbi Decoder, Viterbi Algorithm, Verilog HDL, Soft Decision Decoding, Trace Back Method.*

1. Introduction

In the recent past, the error-correcting coding has become one integral part in nearly all the modern data communication and storage systems. With the continuously increasing demands for higher speed and lower power communication systems, enhanced VLSI implementations of those error-correcting codes that are currently used in practical applications have great current importance. Coding techniques are essential for a communication system to achieve high performance. One of the most important and direct applications of information theory is coding theory. The purpose of Forward Error Correction (FEC) is to improve the capacity of channel by adding some carefully designed redundant information to the data which is being transmitted through the channel. The process of adding this redundant information is known as channel coding. Convolution coding and block coding are two major forms of channel coding. Block codes operate on relatively large message blocks. Convolution codes operate on serial data, one or a few bits at a time.

Convolution codes were invented in 1955 by P.Elias. Convolution codes are generally error correcting codes that are used to improve the performance of many digital systems such as digital radio, mobile phones and the Bluetooth implementations. Viterbi decoder together with its improved versions is one of the best applications of convolutional codes. When a signal is transmitted over the channel, it is affected by any of three channel impairments i.e. noise, interference, fading. To reduce the number of bit errors in the information is accomplished by introducing redundant bits into the transmitted information stream. These bits will allow detection and correction of received data, and provide reliable transmission of information. Convolution coding with Viterbi decoding is a FEC technique that is particularly suited to a channel in which transmitted signal is corrupted mainly by additive white Gaussian noise (AWGN). In most of real time applications like audio and video applications, the convolutional codes are used for error correction. Convolutional code definition parameters are the following: code rate (r), generating polynomial g (n), and number of input bits (k), number of output bits (n) and constraint length (K).

There are three alternative methods that are often used to describe the convolutional code. These are the tree diagram, state diagram and trellis diagram. There exist four basic convolutional codes decoding techniques: sequential, threshold, maximal-likelihood and the Viterbi algorithm. The sequential algorithm can provide very strong correcting capabilities while it needs relatively large memory, which strongly depends on communication

channel error density. The threshold algorithm is extensively good for channels with mid to good signal to noise ratios (SNR). The Viterbi algorithm is an optimum decoding technique. It is optimum as it results in the minimum probability of error. It is also the relatively straight algorithm to implement in hardware and is the best decoding technique. Viterbi algorithm is a maximum likelihood algorithm and performs decoding, through searching the minimum cost path in a weighted oriented graph, called trellis. The basic building blocks of Viterbi decoder are branch metric unit (BMU), path metric unit (PMU), add compare and select unit (ACSU) and survivor memory management unit (SMU).

2. Literature Survey

Feygin et al (1993) compares various memory organization techniques in Viterbi decoder where register exchange method is straight forward and simple. Trace back methods are of four types- k-pointer even algorithm, k- pointer odd algorithm, one point algorithm and hybrid algorithm. Memory size, latency and implementation complexity of the survivor sequence management were analyzed for both uni processor and multiprocessor realization of Viterbi decoder. They confirmed that one pointer trace back was better than trace back methods [1].

Black et al (1993) derived a 140-Mb/s-state, radix-4 Viterbi decoder. The throughput of a radix-2 state parallel Viterbi decoder could be increased by using higher radix formulations of the trellis. Implementation of trace-back algorithm using pre-traced decision was possible without fragmenting the decision memory into multiple memories and without running multiple trace-back recursions in parallel. By using these entire concept, achieved the iteration rate of 70MHz, decoding rate of 140Mb/s was achieved [2].

Berrou et al (1993) analyzed a class of convolution codes, called turbo codes, whose performance is in terms of BER are close to the Shannon limit. Turbo-code encoder is built by parallel concatenation of two RSC codes and an associated decoder. The decoding complexity of the MAP algorithm has been reduced in log-MAP algorithm by operating it in the log domain [3].

Andrew J. Viterbi (1998) presented a shortcut to understanding the maximum a posterior decoder based on an approximation. This corresponds to dual-maxima computation combined with forward and backward recursions of Viterbi algorithm computations. [4].

Hsu et al (1998) presented a parallel decoding scheme for turbo codes. They presented a proposed method for reducing the decoding delay by means of segmenting a block into several sub-blocks, which were partially overlapped. The proposed sub block segmentation scheme allows for the parallel decoding of each component code by using several sub-block decoders. [5]

Curt Schurgers et al (1999) developed Energy efficient data transfer and storage organization for MAP turbo decoder**.** Storage bottleneck in data dominated algorithm (MAP turbo decoding) had reduced by Data Transfer and Storage Exploration (DTSE) through the steps pre processing, global dataflow transformation, global loop and re-indexing. [**6].**

Alexander worm et al (2000) introduced a technique in which the frame could be tiled in a number of windows for high speed architecture. All windows of a frame were processed in parallel as forward and backward recursions started simultaneously at different trellis steps which avoided timing problems at path metric exchange between different windows. In X-RSC architecture, 20% power reduction and 18% area reduction had been achieved [7].

Cheng et al (2000) discussed Linearly Approximated Log-MAP Algorithms for Turbo Decoding. A theoretical framework was proposed to explain why the Log-MAP turbo decoders are more tolerant of over estimation of SNR than underestimation. Simulations indicate that this Log-Lin algorithm achieved the same performance as the Log-MAP algorithm, and that a fixed-point implementation of the Log-Lin algorithm with optimized clipping and quantization to 6 bits, results in a loss of performance of less than 0.05 dB [8].

Wang et al (2002) implemented the sliding window technique, a preliminary backward (forward) recursion unit which started its recursion by initializing each state of equi-probable and passed the resultant state metric after a considerable number of trellises. This has overcome the drawback of MAP algorithm. Instead of replicating the hardware of a serial decoding architecture as in traditional parallel processing, a variety of novel area efficient parallel turbo decoding schemes were presented. In Segmented Sliding Window (SSW) technique, the decoding frame was divided into many sliding blocks and assigned to several segments. The main drawback of SSW was that the pre computation part of real computation ratio of backward state metric was 1:1 which had to be reduced to achieve high area efficiency. [9]

Chien et al (2002) demonstrated BCJR algorithm. According to the computational complexity of the employed decoding algorithm, the realization of turbo decoders usually takes a large amount of memory spaces and potentially long decoding delay. They focused on the development of general formulas for efficient memory management of turbo decoders employing the sliding-window BCJR algorithm. Three simple but general results were presented to evaluate the required memory size, throughput rate, and latency based on the speed and the number of adopted processors [10**].**

Mansour et al (2003) discussed VLSI Architectures for SISO-APP Decoders. This provides an analysis of the requirements for computational hardware and memory at the architectural level based on a tile-graph approach that models the resource-time scheduling of the recursions of the algorithm. The proposed tiling scheme of the recursion patterns, called hybrid tiling, was shown to be particularly effective in reducing memory overhead of high-speed SISO-APP architectures [11].

Indrajit et al (2003) derived Low power VLSI implementation of the MAP decoder for turbo codes through forward recursive calculation of reverse state metrics. The MAP decoder operating in the log domain required either forward or the backward path metrics to be stored before finally calculating the log-likelihood decisions. There is 88% reduction in memory size. This reduction leads to approximately 35% saving in power when compared to the traditional Log- MAP decoder. Power reduction of 17% is also achieved in the RSMC block [12].

Engling Yeo et al (2003) compared four different structures for the implementations of the ACS recursion. These inferences are applicable to the implementations of both soft and hard - decision Viterbi decoders. It was found that architectural retiming and transformation of the ACS structures with modification of the register exchange provided the highest throughput without excessive area and power penalties. Although the SOVA has less complexity than the MAP decoder, it still has higher power consumption than the hard- output Viterbi decoder. In practical high performance iterative decoders, the power could be lowered through custom circuit design and technology scaling [13].

Tiwari et al (2004) presented memory sub-banking scheme for high throughput turbo decoder. Sub-banked implementation of SW-banked approach achieved high throughput. The Sliding Window (SW) approach had been proposed as an effective means of reducing the decoding delay as well as the memory requirements of turbo implementations. They presented a sub banked implementation of the SW-based approach that achieved high throughput, low decoding latency and reduced memory energy consumption. [14].

Tsung-Han Tsai et al (2005) presented a memory-reduced Log-MAP kernel for turbo decoders. In this, backward recursion state vectors were traced back with the forward recursions. They proposed a new log-MAP kernel to reduce memory usage. Proposed architecture could reduce the memory size to 26% of the classical architecture. They simplified the memory data access in this kernel design without extra address generators. For the 3GPP standard, a prototyping chip of the turbo decoder was implemented to verify the proposed memory-reduced log-MAP kernel in 3.04×3.04mm2 core area in the UMC 0.18 m CMOS process [15].

Seok-Jun-Lee et al (2005) combined BIP, a high throughput technique, with the look ahead computation and constructed BIPMAP decoder architecture which provides a throughput gain of 1.96 at the cost of 63% area overhead. Compared to the parallel architecture, the BIP architecture provided the same speed-up with a reduction in logic complexity by a factor of M, where M is the level of parallelism. Reducing memory access is crucial to achieve low power design in the memory intensive algorithm like turbo decoding. The parallel and symbol based decoder architectures improves the throughput at the expense of increased area. The BIP reduces this area penalty without sacrificing the throughput gain. [16]

Russell Tessier et al (2005) described a power-efficient implementation of an adaptive Viterbi decoder. To measure its power consumption, the adaptive Viterbi algorithm (AVA) architecture has been implemented in two contemporary FPGA architectures for a range of constraint lengths. For a given fixed BER and decode rate, power savings was achieved by adapting the constraint length of the convolutional code employed, with the goal of employing a lower power decoder when allowable. The dynamically reconfigurable FPGA implementation was shown to consume significantly less power than a static FPGA implementation [17].

Yun et al (2006) described a novel low - power design methodology of Viterbi decoder. Based on the described SMU management scheme, the number of memory access could be significantly reduced, that could help to reduce the power consumption. The studied SMU management could also be regarded as the trace-back technique with the dynamic survivor length. The salient feature of the

proposed architecture is that the survivor memory can be implemented mainly by only three banks of single-port [18].

Dong-Soo Lee et al (2006) implemented approximate reverse calculation method for backward metrics with simple arithmetic operations such as addition and comparison .They achieved the access rate of the backward metric memory by 87% in W-CDMA standard without affecting error correcting performance. 29% power consumption was reduced in a log- MAP decoder by employing the approximate reverse calculation method. In this paper only 13% of backward metrics values are stored, remaining are not saved because they could be recovered by the reverse calculation. [19].

Arun et al (2007) minimized memory architecture for low latency Viterbi decoder using Zig-Zag algorithm [20]. In forward backward algorithm, when employed in sliding window technique, the time slot is properly utilized to reduce the memory elements. The data storage takes place always in the forward manner and the trace back in reverse manner. Whereas in the Zig-Zag algorithm, storage takes place from right to left in the forward direction and trace back from left to right in the reverse direction but it is exactly opposite in the second time and vice-versa which reduces the memory elements which is not possible in the forward-backward algorithm. The proposed method provided greater area advantage compared to the two RAM trace back method. They had achieved a latency of 50% compared to 100% for the trace back algorithm. Only 56.09% of total memory had been utilized in the proposed algorithm than the trace back method. The operating frequency for the trace back algorithm was 83.243 MHZ, but it is 452.694 MHZ for the proposed approach. Total delay was only 6.785ns but it was 12.013ns for the trace back algorithm approach.

Cheng et al (2009) presented low-power memory-reduced trace back MAP decoding for double-binary convolutional turbo decoder. Power reduction of state metrics cache was achieved by introducing memory reduced trace back MAP decoding technique. In contrast to the conventional MAP decoding the trace back MAP decoding reduces the number of stored metrics by accessing the different metrics. Trace back MAP decoding performed without losing correction ability.. SB MAP decoding required half the SMC size of the conventional decoding procedure. Two trace back structures achieved 20% power reduction of the SMC [21].

Yang Liu et al (2009) discussed Design of voltage over scaled low-power trellis decoders in presence of process variations. Unequal error tolerance involved two main issues i.e. how to quantify the importance of each circuit signal and how to incorporate the importance qualification into signal processing circuit designs [22].

Guan et al (2009) presented various decoding techniques, such as the Log-MAP, Max-log-MAP and SOVA algorithm [23] for non-real-time service were compared in this paper. Among this Log-MAP algorithm was shown to achieve the best performance with good complexity trade off.

Fan yang et al (2010) explained Design of turbo-like codes for short frames. Generally Turbo code performed well for long block sizes. Turbo-like code suggested for shorter frame code and it could be decoded by belief propagation (BP) algorithms [24] with low complexity. They proposed a type of Turbo-Like (TL) codes for short frame transmission. The component encoder of a TL code is a Systematic Feed-Forward Convolutional (SFC) code designed without 4-cycles and the associated component decoder used serial belief propagation rule.

Jinjin He et al (2010) gave memory-reduced MAP decoding for double-binary convolutional turbo code and partitioned the branch metric (BM) without introducing any computational overhead and also extrinsic metrics were independent from a posterior LLR to minimize the memory elements. They presented a memory-reduced VLSI architecture for the decoding of Double-Binary Convolutional Turbo Code (DB CTC) using Maximum a Posteriori Probability (MAP) algorithm [25]. They proposed to decompose each BM into information metric and a parity metric, which leads to 50% reduction of the memory size for BMs. Modified the MAP algorithm based on the new formulation of BMs.

Chavel et al (2010) discussed a memory mapping approach for parallel interleaver design with multiples read and writes access Memory mapping technology **[26]** was applied in parallel interleaver architecture with multiple read/write access.

Yiming Chen et al (2010) developed Iterative soft decision feedback Zig-Zag equalizer for 2D intersymbol interference channels. Zig-Zag algorithm exchanges soft information between MAP decoders. They presented a novel iterative soft decision feedback Zig-Zag algorithm **[27]** for detection of binary images corrupted by two dimensional intersymbol interference and additive white Gaussian noise. Memory arrangements in turbo decoders use sliding-window.

Moncayo et al (2011) discussed the Performance evaluation of a turbo codec with Log-MAP algorithm on FPGA **[28]**. They evaluated the performance of the MAP decoder using FPGA based and software based implementation. Results showed that the FPGA-based Turbo Codec was able to estimate correctly the information sequence; therefore its BER performance was comparable to the software based implementation. However, the results of the comparison also showed that the FPGA-based system was faster and consumes less energy that the software-based system.

Shrestha et al (2011) conferred hardware implementation of Max-Log-MAP algorithm based on MacLaurin series [29] for turbo decoder. Generally hardware implementation of MAP decoder was quite complex. The original MAP algorithm suffers from serious drawbacks in its hardware implementation. To overcome this disadvantage, Max-Log-MAP and Log-MAP algorithms had been proposed to reduce the complexity.

Martina et al (2011) derived the state metric compression techniques for turbo decoder architecture. Two different techniques were applied to compress state metrics in turbo decoder. Non uniform quantization technique [30] reduced state metric memory area; it could be employed with codes that exhibit very low error floor. Walsh – Hadamard transform was well suited to reduce the decoder area when the code error floor should be preserved. Both the techniques mitigate the power consumption of the decoder.

Oletu et al (2011) revealed the performance of turbo codes for wireless communication systems. Soft output Viterbi algorithm and logarithmic - maximum a posteriori turbo decoding algorithm performance were compared [31] and the results shows that Log-MAP had a better BER over SOVA (BER 10-4, K = 3). Soft-output Viterbi algorithm and logarithmic maximum a posteriori turbo decoding algorithms were the two candidates for decoding turbo codes. Soft-Input Soft-Output (SISO) turbo decoder based on Soft-Output Viterbi Algorithm (SOVA) and the logarithmic versions of the MAP algorithm, namely, Log-MAP decoding algorithm.

Karim et al (2011) developed Design of pipelined parallel turbo decoder using contention free interleaver**.** Pipeline technique [32] applied to reduce the critical path delay of the Add Compare Select Offset (ACSO) unit so as to increase the operating clock frequency. Highly parallel turbo decoder required a high bandwidth memory access to satisfy the throughput requirements. The interleaver memory was divided into smaller memory blocks. During the sub block clock cycles the row address were not changed.

Mandwale et al (2015) compared four different approaches **[33].**

Method I: GDIL Technique: GDIL technique is introduced for low area consumption and for reducing delay of the circuit. By comparing GDIL with other traditional CMOS and various pass-transistor logic design techniques less area consumption for GDIL is achieved. With the GDIL approach we can design the wide range of complex logic functions using minimum transistors. This method gives us fast speed, low-power consumption circuits, using minimum number of transistors as compared to other techniques like CMOS and Pass transistor logic design.

Method II: Register Exchange: This is mainly concentrated on light-weighted pipelined serial Viterbi Decoder design for resource saving purpose. The trace back bits are stored in RAM unit instead of register array and TBU is not needed. Also for string the BMU and ACS values 'Metric RAM' is used.

Method III: Adaptive Viterbi Algorithm: Instead of saving data in RAM, data is stored in buffers for designing path storage block. For reducing the time when performing branch metric calculation parallel processing is done with Hamming distance. While designing the ACS unit, ROM and 3bit to 12bit shift register is considered.

Method IV: Non Polynomial Approach: A low probability of error is achieved in this Viterbi decoder paper using non polynomial approach. For BMU, it compares the received bits with expected bits. While designing the ACS unit, the addition of path metric and then subtraction is done for purpose.

Table 1 - Comparative Analysis of various Coding techniques

AUTHOR	YEAR	METHODOLOGY	REMARKS
Mandwale et al	2015	➢ GDIL Technique ➢ Register Exchange ➢ Adaptive Viterbi Algorithm ➢ Non Polynomial Approach	A low probability of error is achieved in this Viterbi decoder paper using non polynomial approach
Karim et al	2011	Pipeline technique	Reduce the critical path delay and increase the operating clock frequency
Oletu et al	2011	SOVA & log- MAP algorithm	Log-MAP had a better BER over SOVA
Martina et al	2011	State metric compression technique	Reduce the decoder area and mitigate the power consumption of the decoder
Shreshta et al	2011	Max-Log-MAP algorithm based on MacLaurin series	Proposed to reduce the complexity but hardware-implementation is difficult
Moncayo et al	2011	Performance evaluation of Log-MAP algorithm on FPGA and CPU	FPGA based Log MAP algorithm is better than CPU based algorithm
Yiming Chen et al	2011	Zig-Zag algorithm	Used in detection of binary images corrupted by two dimensional ISI & AWGN
Chavel et al	2011	Memory mapping technology	High throughput and finds a collision-free mapping of the variables in the memory bank.
Jinjin He et al	2010	Modified MAP algorithm	50% reduction of the memory size and memory-reduced VLSI architecture
Fan Yang et al	2010	Belief propagation (BP) algorithm	Memory-reduced MAP decoding for double-binary convolutional turbo code.
Guan et al	2009	Comparison of Log-MAP, Max-log-MAP and SOVA	Decoding complexity is high for Max log & SOVA. Log-MAP algorithm achieves the best performance
Yang Liu et al	2009	Low-power trellis code algorithm	Low power consumption but how to quantify the importance of each circuit signal was unknown
Cheng et al	2009	Memory reduced trace back MAP decoding technique	Reduces the number of stored metrics by accessing the different metrics, decoding performed without losing correction ability and 20% power reduction
Arun et al	2007	Zig-Zag algorithm	Achieved a latency of 50% compared to 100% for the Trace back algorithm. Zig-Zag algorithm is found to be better than trace back algorithm.
D S Lee et al	2006	Reverse calculation method	Power consumption is still a major issue to be solved.
Yun et al	2006	Viterbi algorithm based on SMU management scheme	The proposed SMU management method can regarded as the trace-back technique with the dynamic survivor length
Russel Tesser et al	2005	Adaptive Viterbi algorithm	Reduced-complexity, reduced decoder power consumption and good noise tolerance
Seok Jun Lee et al	2005	Block-interleaved pipelining (BIP) technique	The MAP decoder is recursive & complex, makes high-speed implementation extremely difficult to realize.
Tsung Han Tsai et al	2005	Memory-reduced Log-MAP algorithm	Proposed architecture could reduce the memory size to 26% of the classical architecture

Tiwari et al	2004	The Sliding Window Approach	High throughput and reduces the decoding delay
Engling Yeo et al	2003	Soft Output Viterbi Algorithm	High throughput and information rates that are very close to the Shannon limit
Indrajit et al	2003	More reduced complexity version of MAP algorithm	The proposed design reduces the memory size & hence reduces the power consumption by35%.
Mansour et al	2003	SISO-APP algorithm	Error correction capability is high, Storage limitation, Savings in area and power in the range of 4.2%–53.1% over existing techniques
Chien et al	2002	Sliding-Window BCJR Algorithm	Uses reduced memory span, Trellis termination of code is not required
Wang et al	2002	Segmented sliding window technique Area efficient parallel decoding Hybrid parallel decoding schemes	Hybrid parallel decoding schemes have been shown to be the best choices for very high level parallelism cases. Increase the decoding throughput and area-efficient
Cheng et al	2000	Log-Lin algorithm	Log-Lin algorithm achieves the same performance as the Log-MAP algorithm
Alexander et al	2000	MAP Algorithm	Optimized memory size and power consumption**,** Less area & power consumption by 18%& 20%. Decoding is complex
Schurgers et al	1999	Data transfer and storage exploration	Reduce decoding delay, Reduction in area and Less energy consumption
Hsu et al	1998	Parallel decoding scheme	Optimizations of the original MAP turbo decoder algorithms are implemented
A J Viterbi et al	1998	Dual maxima MAP algorithm	Reduce decoding delay and Reduce number of computation steps
Berrou et al	1993	Log-MAP algorithm	High throughput and better error correcting capability is obtained.
Black et al	1993	Trace back algorithm	High throughput, Better iteration rate, Better decoding rate and Ideal linear scaling. Implementation is possible without fragmenting the decision memory into multiple memories
Feygin et al	1993	One pointer Trace back method	Used in high speed and large constraint length decoders, low bandwidth &area requirements, few memory banks, low latency and simple circuitry. For a high speed VD the TBA is advantageous as compared to the RE method.

3. Convolution Encoder

An encoder includes extra information in the transmitted signal to reduce the probability of errors in the received signal that may be corrupted by noise. Every two bits of data stream are encoded into three bits for transmission. The ratio of input to output information in an encoder is the **rate** of the encoder; this is a rate 2/3 encoder. The following equations relate the three encoder output bits (Yn2, Yn1, and Yn0) to the two encoder input bits (Xn1 and Xn0) at a time Nt.

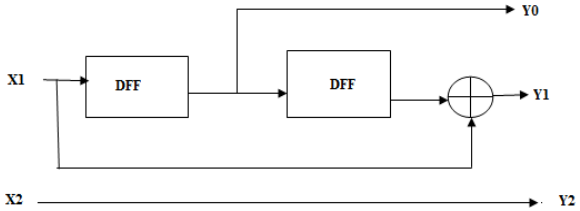

Figure 1. Convolution encoder of rate 2/3

$$Y2 = X2 \quad (1.1)$$

$$Y1 = X1 \wedge Df2 \quad (1.2)$$

$$Y0 = Df1 \quad (1.3)$$

The input bits can be written as a single number. Equation defines a state machine with two memory elements for the two last input values for Xn0 : Df1and Df2 . These two state variables define four states: $\{X_{n-1}^{1}$, Df2 } , with $S_0 = \{ 0, 0\}$, $S_1 = \{1, 0\}$, $S_2 = \{0, 1\}$, and $S_3 = \{1, 1\}$. This model uses two D flip-flops as the state register. When reset (using active-high input signal `res`) the encoder starts in state S_0.

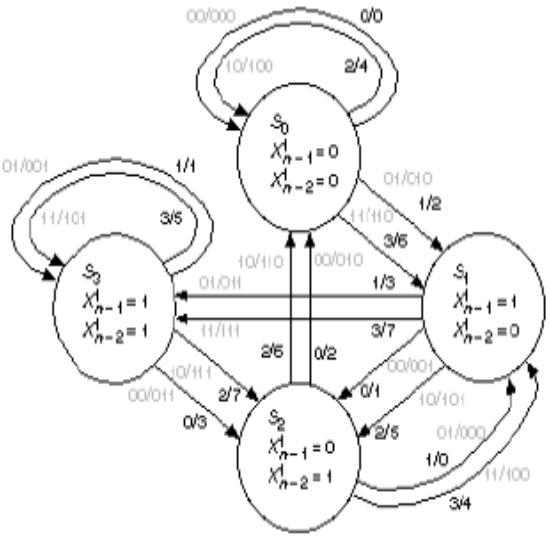

Figure 2. State diagram of a rate 2/3 encoder

4. Viterbi Decoder

The Viterbi decoder consists of three main units branch matrices unit, add control select unit and survivor management unit. Figure shown below shows the general structure of Viterbi decoder.

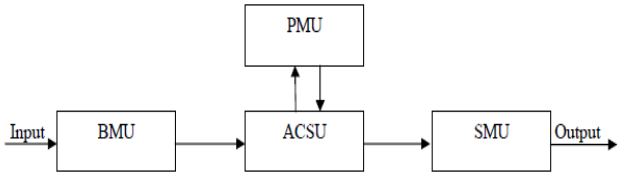

Figure 3. Block diagram of Viterbi decoder

The incoming bit causes transition from one state to another. For a better representation, the state diagram can be unfolded in time to represent transitions at each stage in time. Such a representation is called trellis diagram. States of the trellis diagram represent states of the encoder. Each incoming bit causes trellis to move to next stage. A line joining the two states represents the transition the state in next stage. With each transition there is an associated symbol, called transition symbol, which represents the encoder output for that state and the incoming bits. The branch metric unit (BMU) is responsible for the computation of matrices, second block add compare select unit (ACSU) selects the survivor paths for each trellis state, third block survivor management unit (SMU) performs the selection of output which is based on the most optimum path metric. There are four major steps involved in Viterbi decoding algorithm.

1. Calculate the trellis, which means obtaining the weight of the trellis branches by calculating branch matrices.
2. Obtaining the last state with the minimum path.
3. Trace back which means calculating the entire weight path.
4. Reordering the bits into correct format.

Factors to be focused carefully while designing an efficient error control coding scheme are

- Error detection capability
- Error correction capability
- Encoding complexity
- Decoding complexity

4.1 Branch Metric Unit

The first unit is called branch metric unit. Here the received data symbols are compared to the ideal outputs of the encoder from the transmitter and branch metric is calculated. Hamming distance or the Euclidean distance is used for branch metric computation.

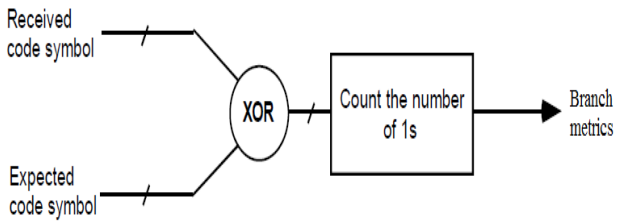

Figure 4. Branch metric computation block

4.2 Path Metric Unit

The second unit, called path metric computation unit, calculates the path metrics of a stage by adding the branch metrics, associated with a received symbol, to the path metrics from the previous stage of the trellis.

4.3 Add Compare Select Unit

The two adders compute the partial path metric of each branch, the comparator compares the two partial metrics, and the selector selects an appropriate branch. The new partial path metric updates the state metric of state *p*, and the survivor path-recording block records the survivor path.

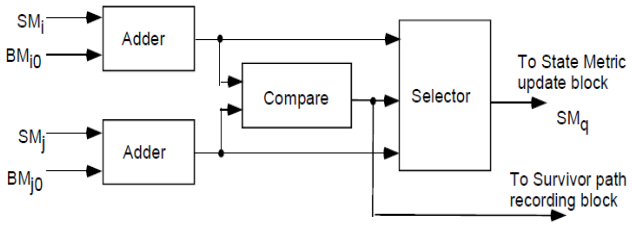

Figure 5. Add Compare Select Unit

4.4 Survivor Management Unit

The final unit is the trace-back process or register exchange method, where the survivor path and the output data are identified. The trace-back (TB) and the register-exchange (RE) methods are the two major techniques used for the path history management in the chip designs of Viterbi decoders. The TB method takes up less area but requires much more time as compared to RE method because it needs to search or trace the survivor path back sequentially. Also, extra hardware is required to reverse the decoded bit.

4.5 Trace Back Method

In the TB method, the storage can be implemented as RAM and is called the path memory. Comparisons in the ACS unit and not the actual survivors are stored. After at least L branches have been processed, the trellis connections are recalled in the reverse order and the path is traced back through the trellis diagram. The TB method extracts the decoded bits, beginning from the state with the minimum PM. Beginning at this state and tracing backward in time by following the survivor path, which originally contributed to the current PM, a unique path is identified. While tracing back through the trellis, the decoded output sequence, corresponding to the traced branches, is generated in the reverse order.

4.6 Register Exchange Method

The register exchange (RE) method is the simplest conceptually and a commonly used technique. Because of the large power consumption and large area required in VLSI implementations of the RE method, the trace back method (TB) method is the preferred method in the design of large constraint length, high performance Viterbi decoders. In the register exchange, a register assigned to each state contains information bits for the survivor path from the initial state to the current state. In fact, the register keeps the partially decoded output sequence along the path, as illustrated in Figure below. The register of state S1 at t=3 contains '101'. This is the decoded output sequence along the hold path from the initial state.

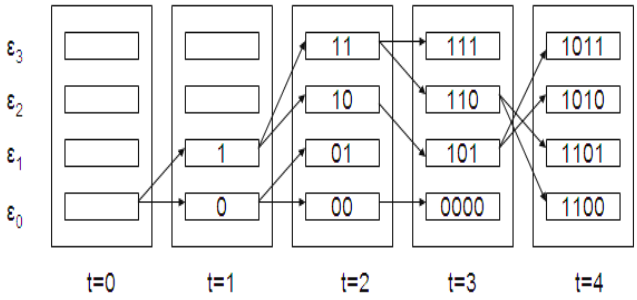

Figure 7 Register Exchange Method

5. Types of Viterbi Decoding

In order to realize a certain coding scheme a suitable measure of similarity or distance metric between two code words is vital. The two important metrics used to measure the distance between two code words are the Hamming distance and Euclidian distance adopted by the decoder depending on the code scheme, required accuracy, channel characteristics and demodulator type.

Hard Decision Viterbi Decoding: In the hard-decision decoding, the path through the trellis is determined using

the Hamming distance measure. Thus, the most optimal path through the trellis is the path with the minimum Hamming distance. The Hamming distance can be defined as a number of bits that are different between the observed symbol at the decoder and the sent symbol from the encoder. Furthermore, the hard decision decoding applies one bit quantization on the received bits.

Soft Decision Viterbi Decoding: Soft-decision decoding is applied for the maximum likelihood decoding, when the data is transmitted over the Gaussian channel. On the contrary to the hard decision decoding, the soft-decision decoding uses multi-bit quantization for the received bits. Euclidean distance as a distance measured instead of the hamming distance. The demodulator input is now an analog waveform and is usually quantized into different levels in order to help the decoder decide more easily. A 3-bit quantization results in an 8-ary output.

6. Simulation Results

Convolution encoder of rate 2/3 and Viterbi decoder is simulated using model Sim. Test benches for encoder and decoder are coded and simulated. Two bits are provided to the encoder as input and three bits are obtained as output of encoder. These encoded three bits are provided as inputs to the Viterbi decoder. Viterbi decoder consists of different modules such as Euclidean distance calculation module, Subset decode module, Compute metric module, Compare Select Module, Path module, Path memory module, Path in Module, Metric Module, Output decision module and reduce module. All these modules are internally connected by Verilog HDL codes and simulated using Model Sim. The output of the Viterbi decoder will be two bit output. When Input to the Convolution encoder is compared to the output of Viterbi decoder, results shows that they are the same.

7. Conclusion

The aim was the construction and design of a convolutional encoder with a Viterbi decoder that can encode a bit stream of digital information and outputs a codeword that has a capability to be transmitted to the destination and then decoded. The encoder was designed with rate 2/3. The Viterbi decoder design had been driven in such a way that it would calculate the decoding path with the minimum metric to be passed to the decoder output port. The trace back method used to decode data from metrics stored in a 16-bit decoding window to generate the decoded output. The decoder has a capability of detecting any error occurs while transmitting over the channel.

References

[1] Feygin, G. and Gulak, P.G. “Architectural tradeoffs for survivor sequence memory management in Viterbi decoders”, IEEE Transactions on Communications, Vol.41, No.3, pp.425-429, 1993

[2] Peter, J. Black and Teresa, H . Meng, “A 140-Mb/s, 32-state. Radix4 Viterbi Decoder”, in: Proceedings of the IEEE Journal of Solid State Circuits, Vol.27 , No.12, pp.1877- 1885, 1993

[3] Berrou, C., Glavieux, A. and Thitimajshima, P. “Near Shannon limit error-correcting coding and decoding: turbo codes (1)”, in: Proceedings of the IEEE International Conference on Communications (ICC), pp.1064-1070, 1993

[4] Andrew, J. Viterbi. “An Intuitive Justification and a Simplified Implementation of the MAP Decoder for Convolutional Codes”, IEEE journal on selected areas in communications, Vol. 16, No. 2, pp. 260- 264,1998.

[5] Hsu, J-H. and Wang, C- L.“A parallel decoding scheme for turbo codes”, IEEE International symposium on circuits and systems (ISCAS), Vol. 4, pp, 445-448, 1998.

[6] Schurgers, C., Catthoor, F. and Engels, M. “Energy efficient data transfer and storage organization for a MAP turbo decoder module”, in: Proceedings of the IEEE International Symposium on Low Power Electronics and Design (ISLPED), pp.76-81, 1999.

[7] Worm, A., Lamm, H. and Wehn, N. “A high-speed MAP architecture with optimized memory size and power consumption”, in: Proceedings of the IEEE workshop on Signal Processing Systems (SiPS), pp. 265- 274, 2000.

[8] Cheng, J-F. and Ottosson, T. “Linearly Approximated Log-MAP Algorithms for Turbo Decoding”, IEEE 51st Vehicular Technology Conference Proceedings (VTC) 2000, Vol. 3, pp. 2252-2256, 2000

[9] Wang, Z., Chi, Z. and Parhi, K.K. “Area-efficient high-speed decoding schemes for turbo decoders”, IEEE Transactions on Very Large Scale Integration (VLSI) systems , Vol.10 , No.6, pp. 902-912, 2002.

[10] Chien, “Memory arrangements in turbo decoders using sliding-window BCJR algorithm”, IEEE International symposium on Circuits and Systems (ISCAS), Vol. 5,pp, 2002.

[11] Mohammad M. Mansour and Naresh R. Shanbhag. “VLSI Architectures for SISO-APP Decoders”, IEEE Transactions on very large scale integration (VLSI) systems, Vol. 11, No. 4, pp 627-650, 2003

[12] Indrajit Atluri and Tughrul Arslan, “Low power VLSimplementation of the MAP decoder for turbo codes through forward recursive calculation of reverse state metrics”, IEEE International system-onchip conference, pp. 408-411, 2003.

[13] Engling yeo, Augsburger, S.A., Rhett Davis, W. and Borivoje Nikolic . “ A 500-Mb/s soft-output Viterbi decoder”, IEEE Journal of solid-state circuits ,Vol. 38, No. 7, pp. 1234-1241,2003.

[14] Tiwari, M., Zhu, Y. and Chakrabarti, C., "Memory sub-banking scheme for high throughput turbo decoder", in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp.29-32, 2004

[15] Tsai, T.H., Lin, C.H. and Wu, A.Y. "A memory-reduced log-MAP kernel for turbo decoder", in: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), pp.1032-1035, 2005

[16] Lee, S.J., Shanbhag, N.R. and Singer, A.C. "Area-efficient highthroughput MAP decoder architectures", IEEE Transactions on Very Large Scale Integration (VLSI) systems, Vol.13 , No.8, pp. 921-933, 2005.

[17] Russell Tessier, Sriram Swaminathan, Ramaswamy, Dennis Goeckel and Wayne Burleson. "A Reconfigurable, power-efficient adaptive Viterbi decoder", IEEE Transactions on Very Large Scale Integration (VLSI) systems, Vol. 13, No. 4, pp. 484-488, 2005

[18] Yun-Nan Chang and Yu-Chung Ding. "A lower-power Viterbi decoder design methodology based on dynamic survivor path decision", IEEE International symposium on VLSI design, automation and test, pp. 1-4, 2006

[19] Lee, D.S. and Park, I.C., "Low-power log-MAP decoding based on reduced metric memory access", IEEE Transactions on Circuits and Systems, Vol.53, No.6, pp.1244-1253, 2006

[20] Arun, C. and Rajamani, V. "Minimized memory architecture for low latency Viterbi decoder using Zig-Zag algorithm", International Journal on Wireless and Optical Communications (IJWOC), Vol.4, No.3, pp. 313-323, 2007

[21] Cheng-Hao Tang, Cheng-Chi Wong, Chih-Lung Chen, Chien-Ching Lin and Hsie-Chia Chang. "A 952MS/s Max-Log MAP Decoder Chip using Radix-4 Ã- 4 ACS Architecture", IEEE Asian Solid-State Circuits Conference, pp. 79-82, 2006

[22] Yang Liu, Tong Zhang and Jiang Hu. "Design of voltage over scaled low-power trellis decoders in presence of process variations", IEEE Transactions on Very Large Scale Integration (VLSI) systems, Vol.17 , No.3, pp. 439-443, 2009

[23] Guan, M. and Yang, M. "Comparison and design of decoder in B3G Mobile communication system", Communications and Network, pp. 20-24, 2009

[24] Fan Yang, Zhendong Luo, Baoyu Tian and Yubai Li. "Design of turbo like codes for short frames", IEEE Communications Letters, Vol. 14, No.2, pp.172-174, 2010

[25] Jinjin He, Zhongfeng Wang and Huaping Liu. "Memory-reduced MAP decoding for double-binary convolutional turbo code", in: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), pp. 469-472, 2010

[26] Chavet, C. and Coussy, P. "A memory mapping approach for parallel interleaver design with multiples read and write accesses", IEEE International Symposium on Circuits and Systems ISCAS, pp. 3168-3171, 2010

[27] Yiming Chen. "Iterative soft decision feedback Zig-Zag equalizer for 2D intersymbol interference channels", IEEE Journal on selected areas in communications , Vol. 28, No. 2, pp. 167-180, 2010

[28] Moncayo, H.I.R., Kaabouch, N., Wen-Chen Hu and Anlei Wang. "Performance evaluation of a turbo codec with Log-MAP algorithm on FPGA and CPU", IEEE International Conference on Electro/Information Technology, pp. 1-6, 2011

[29] Shrestha,R. and Paily, R."Hardware implementation of Max-Log-MAP algorithm based on MacLaurin series for turbo decoder", International conference on Communications and Signal Processing (ICCSP), pp.509-511, 2011

[30] Martina, M. and Masera, G. "State Metric Compression Techniques for Turbo Decoder Architectures", IEEE Transactions on Circuits and Systems, Vol.58, No.5, 2011

[31] Oletu, G. and Rapajic, P. "The performance of turbo codes for wireless communication systems", International Conference on Computer Research and Development (ICCRD), pp. 346-349, 2011

[32] Karim, M.U., Khan, M.U.K. and Khawaja, Y.M. "An area reduced, speed optimized implementation of Viterbi decoder", International Conference on Computer Networks and Information Technology (ICCNIT), pp. 93-98, 2011

[33] Mandwale A. J and Mulani A O" Different Approaches For Implementation of Viterbi decoder on reconfigurable platform"International Conference on Pervasive Computing (ICPC) 2015

[34] Text book on Applied Coding and Information Theory for Engineers by Richard B. Wells, Prentice Hall, Information and system science series.

Author Profile

Rakhi B. Menon completed her B.Tech in Electronics & Communication Engineering under Mahatma Gandhi University. Currently she is pursuing M.Tech in Electronics with specialization in VLSI and Embedded System under Cochin University of Science and Technology (CUSAT).

Dr. Gnana Sheela K received her Ph D in Electronics & Communication from Anna University, Chennai. She is working as a Professor, Department of ECE, TOC-H Institute of Science and Technology. She has published 20 international journal papers. She is life member of ISTE.