FPGA Implementation of Viterbi Decoder for Long Survivor Path

Deepali R.Laddha¹, Archana O. Vyas²
M.E. Student¹, Assistant Professor²
Department of Electronics and Telecommunication
G. H. Raisoni College of Engineering & Management, Amravati, India

Abstract:
Viterbi decoder in wireless communication is always required especially demanded for high speed, low power and low cost. In this paper, we propose to design a Viterbi decoder that uses survivor path for wireless communication in an attempt to reduce the power and cost and at the same time increase the speed. The proposed design is coded in VHDL and implemented on FPGA. Proposed design is expected to have more speed since the processing execution time will be tried to reduce. Furthermore, the proposed design is capable of supporting frequency more than 790 MHz for constraint lengths 7, and 9, rate 1/3 and long survivor path more than 4.

Keywords: Convolutional codes, Field Programmable Gate Array (FPGA) Implementation, Hybrid Register Exchange Method, VHDL, Viterbi decoder.

I. INTRODUCTION

Communication systems play a major role in our daily life, for some applications data might be transmitted for long distances which may exposed to noise sources. These effects could cause changes in data values causing data corruption and loss which led to the introduction of channel coding to detect and correct transmitted data. The channel coding is divided into two main types Block codes and Convolutional codes. This classification is based on the presence or absence of memory in the encoders for these two codes. A block code encoder is memory less as it maps symbols into a bigger codeword and the encoder has no “memory” of other previous input symbols where as the output of encoding a convolutional code is determined by the current input convolving with the preceding input symbols. Each input is memorized by the encoder for a certain amount of time so that it affects not only the current output but also the next output code words. Convolutional encoding and decoding is an important type of channel coding that categorized in a special channel coding methods called Forward error correction which has the capability to decode some data before the end of receiving. An advantage of convolutional coding is that it can be applied to a continuous data stream as well as to blocks of data. Like block codes, convolutional codes can be designed to either detect or correct errors but are mainly used for error correction. The Convolution Encoder and the maximum likelihood Viterbi decoder solve the problems stated above, this paper will allow the receiver to detect and correct the error without the need of retransmission. The aim of this paper is to design and construct an encoder and a Viterbi decoder for convolutional codes.

II. CONVOLUTIONAL ENCODER

A convolutional encoder is made of a fixed number of shift registers. Each input bit enters a shift register and the output of the encoder is derived by combining the bits of the shift registers. The number of output bits depends on the number of modulo 2-adders used with the shift registers.

![Figure 1. Block diagram of convolutional encoder](image)

From Figure 1, the operation of the encoder is just XORing (which are the plus symbols in Figure 1) the binary input in order to generate the output stream. There are some parameters on which the encoder depends. Following Table1, shows the parameter of encoder design.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input number</td>
<td>1</td>
</tr>
<tr>
<td>Output number</td>
<td>3</td>
</tr>
<tr>
<td>Rate of the encoder</td>
<td>1/3</td>
</tr>
<tr>
<td>Constraint length of the encoder</td>
<td>9</td>
</tr>
<tr>
<td>Total Number of inputs</td>
<td>12</td>
</tr>
</tbody>
</table>

Now for the encoder having input as 12 bits; we will have the output of 36 bits.
2.1. State Diagram

![State Diagram of Convolutional Encoder](image)

**Figure 2. State diagram of convolutional encoder**

From the above state diagram of convolutional encoder, the following table can be concluded.

<table>
<thead>
<tr>
<th>I/P Data</th>
<th>Current State</th>
<th>O/P Bits</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>000</td>
<td>000</td>
<td>000(a)</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>111</td>
<td>111(b)</td>
</tr>
<tr>
<td>0</td>
<td>010</td>
<td>011</td>
<td>011(c)</td>
</tr>
<tr>
<td>1</td>
<td>110</td>
<td>100</td>
<td>100(d)</td>
</tr>
<tr>
<td>0</td>
<td>001</td>
<td>111</td>
<td>111(b)</td>
</tr>
<tr>
<td>1</td>
<td>101</td>
<td>000</td>
<td>000(a)</td>
</tr>
<tr>
<td>0</td>
<td>011</td>
<td>100</td>
<td>100(d)</td>
</tr>
<tr>
<td>1</td>
<td>111</td>
<td>011</td>
<td>011(c)</td>
</tr>
</tbody>
</table>

**Table 2. Table from state diagram**

III. VITERBI DECODER

The algorithm used to achieve the Viterbi decoder is simplified in the following flow chart.

![Flowchart of Viterbi Decoder](image)

**Figure 3. Flow charts of Viterbi decoder**

The Viterbi decoder is a straightforward implementation of the basic processes of the Viterbi algorithm. The design consists of three functional units, as shown in Figure 4.

**Figure 4. Block diagram of Viterbi decoder**

The three functional units of the Viterbi decoder are as follows:

1. **Branch Metric Unit**: The BMU is the simplest block in the Viterbi decoder design. The operation of BMU is crucial as it is the first stage of the Viterbi algorithm and the decoding process depends on all the information it provides. A branch metric unit is to calculate branch metrics which are the distances between the received symbols pair of bits.

2. **Add Compare Select Unit (ACSU)**: The add compare select unit also known as the path metric unit (PMU) calculates new path metric values and decision values. Because each state can be achieved from two states from the earlier stage, there are two possible path metrics coming to the current state. The path with the better metric is chosen and stored as the new path metric for current state, while generating a decision bit. The decision bit indicates which branch has chosen. The ACSU needs the results from the calculations of the previous steps, it forms a feedback loop with the external memory unit, where the results are stored.

3. **Survivor Path unit (SPU)**: In the decoder, the SPU is the block which recovers the received data based on all the information from the PMU. It also consumes a large amount of power.

**Hybrid Register Exchange Method (HREM)**: HREM is a combination of register exchange method and Traceback method hence the name Hybrid register exchange method. This method reduces the power and switching activity. In HREM instead of processing single bit in a cycle, now two bits are decoded, which reduces the switching activity to half as compared to RE method. In this method we are using a property of trellis. Initial state can be first traced back through an m cycle. Then contents of initial
state transfer to current state and the next m bits of the register is the m bits of current state itself as shown in Figure 6.

**IV. SIMULATION AND RESULT**

Following are the outputs of the Viterbi decoder observed.

**Simulation:**
Input given to the Viterbi system is 000011110000. For this input we have got the minimum metric at 0, hence the output of the Viterbi system is 000011110000 at output0.

**V. COMPARISON TABLE**

Following table gives us the comparison of power, time, frequency and technology used between the referred papers and the proposed design of the Viterbi decoder.

<table>
<thead>
<tr>
<th>Referred Paper No.</th>
<th>Power</th>
<th>Time</th>
<th>Frequency</th>
<th>Technology Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>62mW</td>
<td>1.267ns</td>
<td>799MHz</td>
<td>Adaptive Viterbi</td>
</tr>
<tr>
<td>[5]</td>
<td>68mW</td>
<td>21.27ns</td>
<td>47MHz</td>
<td>Trace back</td>
</tr>
<tr>
<td>[2]</td>
<td>58mW</td>
<td>10ns</td>
<td>100MHz</td>
<td>Modified state exchange</td>
</tr>
<tr>
<td>[9]</td>
<td>125mW</td>
<td>2.29ns</td>
<td>97MHz</td>
<td>Trace back</td>
</tr>
<tr>
<td>[10]</td>
<td>0.34mW</td>
<td>1.906ns</td>
<td>524.576MHz</td>
<td>Parallel processing</td>
</tr>
<tr>
<td>Proposed design</td>
<td>0.02mW</td>
<td>1.200ns</td>
<td>833MHz</td>
<td>Hybrid Register Exchange</td>
</tr>
</tbody>
</table>

**VI. ADVANTAGES**

- The main advantage of Viterbi decoder is its 1 bit error correction capability.
- An advantage of convolutional coding is that it can be applied to a continuous data stream as well as to blocks of data.
- In this we have used Hybrid Register Exchange Method (HREM) which is the combination of Trace Back (TB) method and Register Exchange (RE) method for decoding the bit stream. HREM is having an advantage over TB method and RE method that it reduces the switching activity due to which power can be reduced.
- The design of hybrid tracing system of trace back and trace forward (register exchange) that has the advantages of both i.e. it will effectively increase the performance due to the reduced switching activity.
- The FPGA advantage, for its friendly VHDL language interface and easy debugging.
VII. CONCLUSION

The aim of this paper 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 is designed with code rate 1/3. 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 proposed Viterbi decoder uses Hybrid Register Exchange Method (HREM) technique for decoding and to reduce power consumption. HREM technique reduces the switching activity and results in less memory operations. The decoder has a capability of detecting any error (or consecutive errors within the boundaries of the decoder) occurs while transmitting over the channel. Convolutional encoder and Viterbi decoder design have been successfully implemented on the FPGA kit.

VIII. REFERENCES


