

Implementation of High Speed Viterbi Decoder using FPGA

Amruta J. Mandwale

Electronics & Telecommunication Department
SKN Sinhgad College of Engg., Korti Pandharpur,
India

Altaf O. Mulani

Electronics & Telecommunication Department
SKN Sinhgad College of Engg., Korti,
Pandharpur, India

Abstract— Data transmitted over any communication channel is affected due to noise, attenuation, distortion. For detecting and correcting the errors due to channel noise, encoding and decoding should be performed at the transmitter and receiver end respectively. The Viterbi algorithm is most popular algorithms for decoding convolution codes. Using VLSI technology, we can design the system which requires low power, less area and high speed constraints. A high-speed and low power Viterbi decoder design is a challenging task due to the recursive iteration of various steps followed for decoding process. In this paper, low power design of Viterbi decoder along with their performance has been discussed.

Keywords— Convolutional encoder, Register exchange, Adaptive viterbi decoder, Non-polynomial-approach, Viterbi decoder, VLSI, FPGA.

I. INTRODUCTION

Data transmissions over wireless channels are affected by attenuation, distortion, interference and noise, which affect the receiver's ability to receive correct information. Convolutional encoding with Viterbi decoding is a method for forward error detection and correction. It has been widely deployed in many wireless communication systems to improve the limited capacity of the communication channels.

It is well known that data transmissions over wireless channels are affected by noise, which affect the receiver's ability to receive correct information. Convolutional encoding with Viterbi decoding is a powerful method for forward error detection and correction. It has been widely deployed in many wireless communication systems to improve the limited capacity of the communication channels.

Like any error-correcting code, a convolutional code works by adding some structured redundant information to the user's data and then correcting errors using this information. The encoder structure is shown in Fig.1. Assume input bit is 11011. So by this encoder we get encoded output equal to 11 01 01 00 01.

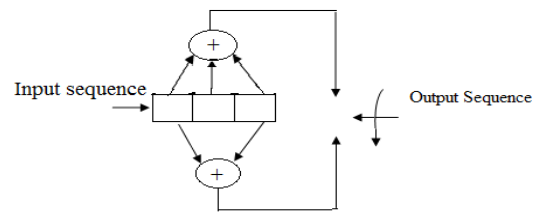


Fig.1: convolution Encoder

Generally, a Viterbi decoder consists of three basic computation units:

- Branch Metric Unit (BMU)
- Add- Compare-Select Unit (ACSU) and
- Trace Back Unit (TBU).

The BMU calculates the branch metrics by the hamming distance or Euclidean distance and the ACSU calculates a summation of the branch metric from the BMU and previous state metrics, which are called the path metrics. After this summation, the value of each state is updated and then the survivor path is chosen by comparing path metrics. The TBU processes the decisions made in the BMU and ACSU and outputs the decoded data. The feedback loop of the ACSU is a major critical path for the Viterbi decoder.

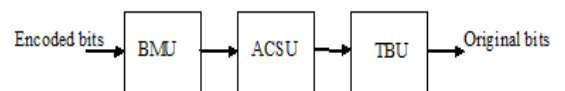


Fig.2: Viterbi decoder data flow

II. FPGA

FPGAs (Field-Programmable Gate Arrays) are PLDs with which can be interconnected to only a few neighbouring cells. A typical FPGA might have 100 cells, each with only 8 inputs and 2 outputs. The output of each cell can be programmed to be an arbitrary function of its inputs. FPGAs typically have a large number (>64) of I/O pins. FPGA architectures vary in the complexity of their individual cells (simple cells are "fine-grained" and complex cells are "coarse-grained") and the flexibility of the interconnections between cells ("routing resources"). Simple cells are arranged as simple ROM-like look-up tables (LUTs) while more complex cells such as the Xilinx CLBs (Configurable Logic Blocks) can include more specialized logic such as carry look-head generators for high speed adders and may include multiple levels of logic within the cell.

III. VITERBI DECODER

Since the received signal is analog, it can be quantized into several levels. If the received signal is converted into two levels, either zero or one, it is called hard decision. If the input signal is quantized and processed for more than two levels, it is called soft decision. The soft decision captures more information in the input signal consequently performing better than the hard decision at the cost of a higher complexity. In this thesis, the ML with the hard decision has been employed.

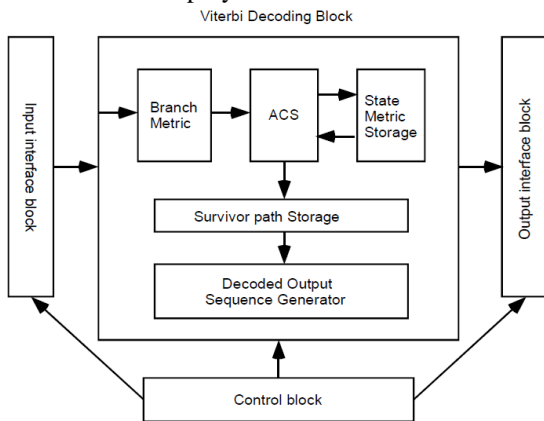


Fig. 3: General Viterbi Decoder

- **Input and output blocks:** Input and output blocks provide the interface with the external components. In the case of radio communications, input received by the decoding block is usually serial, while the decoding block actually needs a parallel input. Serial to parallel conversion and vice versa are carried out by the input and output blocks.
- **Branch Metric:** This block calculates the branch metric of each stage in the trellis. It also calculates the hamming distances (i.e. branch metric) between the received symbol and expected symbol.
- **State Metric Storage:** The block stores the partial path metric of each state at the current stage.
- **ACS:** The Add-Compare-Select block receives two branch metrics and the state metrics. An ACS module adds each incoming branch metric of the state to the corresponding state metric and compares the two results to select a smaller one. The state metric of the state is updated with the selected value, and the survivor path information is recorded in the survivor path storage module.
- **Survivor Path Storage:** The survivor path storage block is necessary only for the trace back approach. The block records the survivor path of each state selected by the ACS module. It requires one bit of memory per state per stage to indicate whether the survivor path is the upper one or the lower one.
- **Output Generator:** This block generates the decoded output sequence. In the trace back approach, the block incorporates combinational logic, which traces back along the survivor path and latches the path (equivalently the decoded output sequence) to a register.

IV. PROPOSED SYSTEM DESIGN

Fig. 4 describes the interfaces of the Viterbi decoder, as well as its internal building block structure. The core uses three AXI4-Stream interfaces (data input, data output, and control). The core allows a simple adaptation for different standards, as well as different requirements.

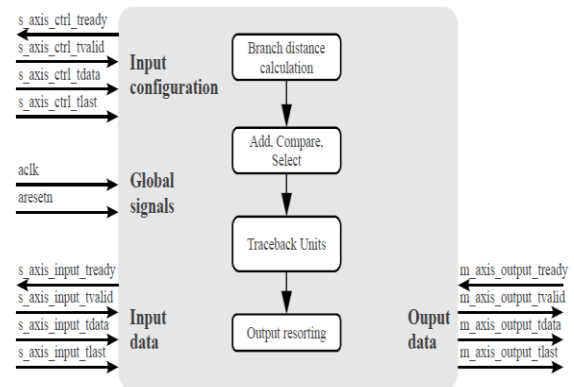


Fig. 4: Block Diagram of Viterbi decoder

- **Branch Distance Calculation (branch_distance.vhd):**
 This part computes all possible branch distances and does not depend on any other values or entity, but the input values. The branch distance is calculated by a Euclidean distance. Thereby the incoming LLR value is added or subtracted to the overall distance. The parity LLR input values for one information bit are read and added or subtracted in parallel. Depending on the code rate, 4 (code rate 1/2), 8 (code rate 1/3), or 16 (code rate 1/4) branch distances exist.
- **Add, Compare and Select (acs.vhd)**
 The add, compare, and select (ACS) entity acs calculates the local most likely path, as described in the Viterbi algorithm. The ACS unit uses a recursion and therefore contains in most cases critical unit of the Viterbi decoder. Based on the Viterbi decoder each node of the trellis tree is represented as one ACS entity.
- **RAM Control (ram_ctrl.vhd)**
 The data produced by the ACS units, needs to be stored for further processing. The ram_ctrl entity is used as glue logic between ACS unit and trace back unit, which is used to perform the second part of the Viterbi algorithm.
- **Traceback (traceback.vhd)**
 The traceback unit is required for the second part of the Viterbi algorithm. From a given starting state the decisions calculated by the ACS unit is processed backwards. As a result of this the output is in a reversed order. The traceback always starts at the zero state of the trellis. Besides the starting state, there is no further information about the decoder configuration applied to the traceback unit.

V. EXPERIMENTAL RESULTS

This section provides some simulation results for Viterbi decoder on Xilinx XC6VLX240T FPGA chip. The ISE 13.1 is used for synthesizing the VHDL modeling codes. The device utilization summary is shown in Table 1. & comparison with previous research work is given in Table 2.

Output of ISE simulator is shown in Fig 5. & overall RTL schematic for the Viterbi decoder is shown in Fig 6.

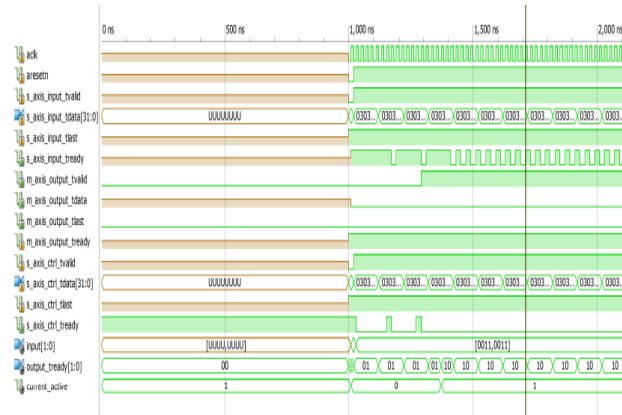


Fig. 5: Top Module Test Bench Waveform for encoder output 03030303

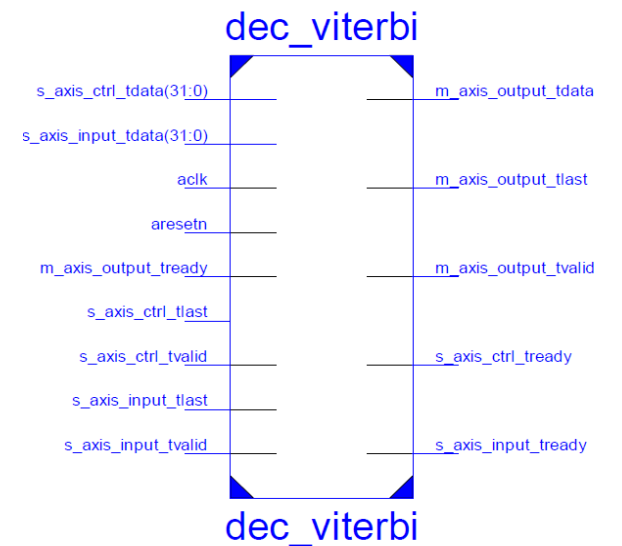


Fig. 6: RTL schematic for the Viterbi decoder

Table 1. Synthesis result

| Sr. No. | Utilized Parameters | Utilized | Available | Optimization |
|---------|----------------------------------|----------|-----------|--------------|
| 1 | Number of Slices | 3542 | 150720 | 2% |
| 2 | Number of Slice Registers | 1318 | 31440 | 0% |
| 3 | Number of fully used LUT-FF pair | 1284 | 3576 | 35% |
| 4 | Number of bonded IOBs | 33 | 600 | 5% |
| 5 | Number of GCLKs | 1 | 32 | 3% |

The project design model achieves good performance and occupies less power. Table 2 shows the previous research work compares with our work. Here we have used the FPGA device as XC6VLX240T with bit rate 1/2 and constraint length of 7 so we have achieved the low power design of Viterbi decoder as shown in Table 2.

Table 2. Synthesis operating frequency comparisons with Previous Research

| Sr.No | Design by | Device | Rate | Constraint length | Operating Frequency (MHz) |
|-------|----------------------|-------------|------|-------------------|---------------------------|
| 1 | Devendra Made et.al | XC3S200 | 1/3 | 7,9 | 790 |
| 2 | Gaurav Swairag et.al | XC3S200 | 1/3 | 3 | 338.52 |
| 3 | K.Cholan et.al | XC3S200 | 1/2 | 3 | 350.294 |
| 4 | Siddeeq et.al | Matlab | 1/3 | 7,9 | 790.266 |
| 5 | Our Work | XC6VLX 240t | 1/2 | 7 | 255.037 |

VI. CONCLUSION

As Viterbi algorithm is conceived more interesting and challenging for any research topic and also it has wide variety of applications in digital communications field. Viterbi decoder has many advantages like low power consumption, high speed, less errors in the system and we have achieve these with less complication using VLSI. Our system utilizes 3542 slices with maximum operating frequency of 255.037 MHz, It operates at a speed of 3.921 ns and which is less as compared to previous research work.

VII. REFERNCES

- [1]. Gaurav S. Wairagade, Surekha K. Tadse, "Design of Asynchronous Viterbi Decoder using VHDL for Low Power Consumption", International Journal of Research in Computer and Communication Technology, Vol 3, Issue 4, April 2014
- [2]. Ye Zhang, Zhimiao Chen, Ralf Wunderlich and Stefan Heinen, "Low-Effort High-Performance Viterbi-based Receiver for Bluetooth LE Applications", IEEE, 2013
- [3]. Devendra Made, Asst. Prof. R.B. Khule, Dipak Iwanate, "Implementation of Adaptive Viterbi Decoder", International Journal of Engineering Trends and Technology-Volume4Issue2- 2013
- [4]. Mr. Sandesh Y.M, Mr. Kasetty Rambabu, "Implementation of Convolution Encoder and Viterbi Decoder for Constraint Length 7 and Bit Rate $\frac{1}{2}$ ", Int. Journal of Engineering Research and Applications, Nov-Dec 2013
- [5]. Rajesh C, A Sreenivasa Murthy, "VHDL Implementation of Fast and Efficient Viterbi decoder", International Journal of Advanced Computer Research, Sep-2013
- [6]. Trang Hoang, Viet Vo Quoc, Truong Nguyen Ly Thien, "FPGA Architecture of HMM-based Decoder Module in Speech Recognizer", International Conference on Control, Automation and Information Sciences (ICCAIS), 2012
- [7]. K.Cholan, a*, "Design and Implementation of Low Power High Speed Viterbi Decoder", International Conference on Communication Technology and System Design, 2011
- [8]. M. Umar Karim, Muhammad U. K. Khan, Yahya M. Khawaja, "An area reduced, speed optimized implementation of Viterbi decoder", IEEE, 2011
- [9]. Prof. Siddeeq Y. Ameen, Mohammed H. Al-Jammas, Ahmed S. Alenezi, "FPGA Implementation of Modified Architecture for Adaptive Viterbi Decoder", IEEE, 2011
- [10]. Anubhuti Khare, Manish Saxena, Jagdish Patel, "FPGA Based Efficient Implementation of Viterbi Decoder", International Journal of Engineering and Advanced Technology (IJEAT), October 2011
- [11]. Takeharu Kohri, "Location Estimation Method Based on Viterbi Algorithm", IEEE, 2009
- [12]. Russell Tessier, Sriram Swaminathan, Ramaswamy Ramaswamy, Dennis Goeckel, and Wayne Bursleson, "A Reconfigurable, Power-Efficient Adaptive Viterbi Decoder", IEEE, April 2005
- [13]. S. L. Haridas and Dr. N. K. Choudhari, "A Low Power Viterbi Decoder Design With minimum Transition Hybrid Register Exchange Processing For Wireless applications", International journal of VLSI design & Communication Systems (VLSICS) Vol.1, No.4, December 2010
- [14]. Zhenzhi Wu, Shujuan Hou, Hai Li, "A light-weighted Viterbi Decoder implemented by FPGA", International Conference on Instrumentation, Measurement, Computer, Communication and Control 2011
- [15]. B. Pullaiah and Dr. M. Sailaja, "VLSI Implementation of Encoder and Decoder for Advanced Communication Systems", International Journal on Recent and Innovation Trends in Computing and Communication October 2014