Design and Implementation of Convolutional Encoder and Parallel Processing Viterbi Decoder Using Verilog

Vinay .B .K[1], Sunil .M .P[2]

[1]M.Tech Student (SP & VLSI) Department of Electronics and Communication Engineering, Jain University, Karnataka, India

[2]Assistant Professor, Department of Electronics and Communication Engineering, Jain University, Karnataka, India

Abstract: ***Convolutional encoding is a forward error correction technique that is used for correction of errors at the receiver end. Viterbi decoding is the best technique for decoding the convolutional codes. Convolution encoder and Viterbi decoder are widely used in many communication systems due to the excellent error control performance. This work deals with the design and implementation of convolution encoder and Viterbi decoder using Field Programmable Gate Array. By analysis the algorithm of Viterbi decoder, the project explores a practical method to design a parallel processing Viterbi decoder. It means trace back and decoder can simultaneously work in order to improve the processing speed. Due to parallelism, Total latency in decoding first bit will be 3L as compared to conventional approach which is 4L, where L is Traceback length. The experimental results show that this method is feasible.***

Keywords: Viterbi, Traceback, Branch metric unit, Add compare select

1. Introduction

The Viterbi decoding algorithm, proposed in 1967 by Viterbi, is a decoding process for convolution codes in memory-less noise. The algorithm can be applied to a host of problems encountered in the design of communication systems. The Viterbi maximum-likelihood and a maximum posteriori algorithm. A maximum posteriori algorithm identifies a code word that maximizes the conditional probability of the decoded code word against the received code word; in contrast a maximum likelihood algorithm identifies a code word that maximizes the conditional probability of the received code word against the decoded code word.

The two algorithms give the same results when the source information has a uniform distribution. A normal Viterbi decoder traces back first and then decode results. The aim of this paper was to design a parallel processing Viterbi decoder. It means that this design can continuously trace back and simultaneously process decoding results. In this way, the Viterbi decoder is more efficient. This has been the focus in the present paper. Several important design issues have also been discussed in the paper, such as organization of convolution encoder and Viterbi decoder, branch metric unit computation method, the design of add compare select, parallel unit of trace back and decoder. A normal Viterbi decoder is usually composed of five blocks, as shown in the Fig. 1.

BMU (Branch Metric Unit) receives the input signals and compute the difference between the input signals and the expected values, and then outputs the results to the Add Compare Select. ACS (Add Compare Select) completes the survivor paths and generates the decision vector. This module has two comparators, and these two comparators are parallel.

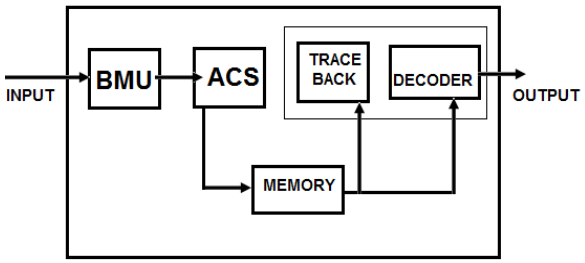


Figure 1: Viterbi Decoder Block Diagram.

The one is used for comparing signal bit and the other is used for comparing data bits. The advantage of this design is considerably reducing the circuit timing delay. The core unit of this design is the TB (Trace Back) and decoder, which include two function modules. The one is TB that read the information of the survivor paths in the memory. The other is decoder that computes the output results. These two modules work simultaneously but in different memory address. In this way, the Viterbi decoder is more efficient than a normal one because the decoder do not need to wait for TB.

This paper is structured as follows. Section 2 introduces the approach to design the convolution encoder. Section 3 formulates the practical and efficient approach to design the Viterbi decoder. At the same time, discuss some important issues related to the Viterbi decoder. Section 4 presents experimental results. Section 5 gives conclusion.

2. Convolutional Encoder

The encoder in Figure 2 produces three bits of encoded information for each bit of input information, so it is called a rate 1/3 encoder. A convolution encoder is generally characterized in (n, k, m) format, where n is number of outputs of the encoder; k is number of inputs of the encoder; m is number of memory elements (flip-flops) of the longest shift register of the encoder. The rate of a (n, k, m) encoder

is register is required. The input data is given in the serial format. Data is shifted with each clock pulse according to the generator polynomials, the inter connections between the shift register and modulo-2 adder are made. Three bit output is produced by modulo-2 addition (i.e. XOR operation) of polynomial.

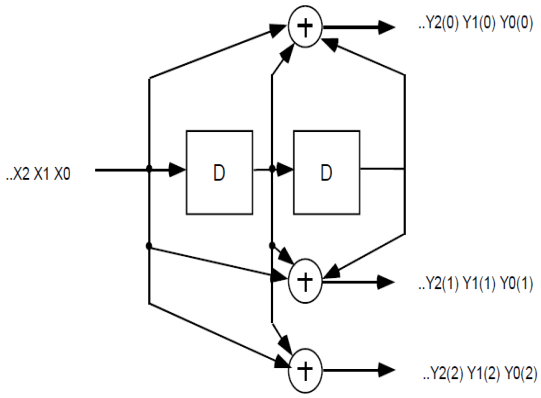


Figure 2: Convolution encoder with generator polynomial (7, 7 and 3)

3. Viterbi Decoder Implementation

A Viterbi decoder is usually composed of three blocks. It consists of following blocks:

- Branch Metric Unit(BMU)
- Add Compare and Select Unit(ACSU)
- Traceback Unit(TU)

Each block is described in the following sub-sections.

A. Branch Metric Unit (BMU)

This unit computes the Branch Metric (BM) for the received symbol. In case of hard decision, BM is taken as the Hamming distance between received symbol and the encoder output corresponding to a particular state transition. In case of soft decision, BM is taken as Euclidean distance. The BMU performs EX-OR operation on received symbols and expected symbols and counts the number of differed bits i,e. number of 1s, by which Hamming distance is obtained. The result is fed as input to the Add Compare and Select Unit. Figure 3 shows Branch metric computation block.

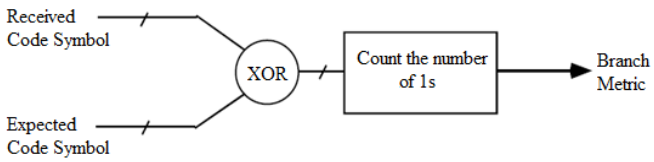


Figure 3: A branch metric computation block

B. Add Compare and Select Unit (ACSU)

The Add Compare Select Unit (ACSU) carries the bulk of arithmetic processing of the Viterbi decoder. ACSU is also called as Path Metric Unit (PMU), as path metric is computed. The main operations involved in the path metric computation are addition, comparison and selection. Figure 4 shows the link between a stage in the trellis diagram and the ACS unit. It can be noted that two ACS Unit that share the same inputs can be grouped in a *Butterfly* Unit. The ACSU unit is composed of 4 ACS units, each is composed of an ACS butterfly module, which adds the corresponding BM to corresponding PM, compares the new PM, feeds the selected PM to ACSU unit and generates the decision bits. The decoder implements adders for computing the path metric and a compare select section to decide on the best path.

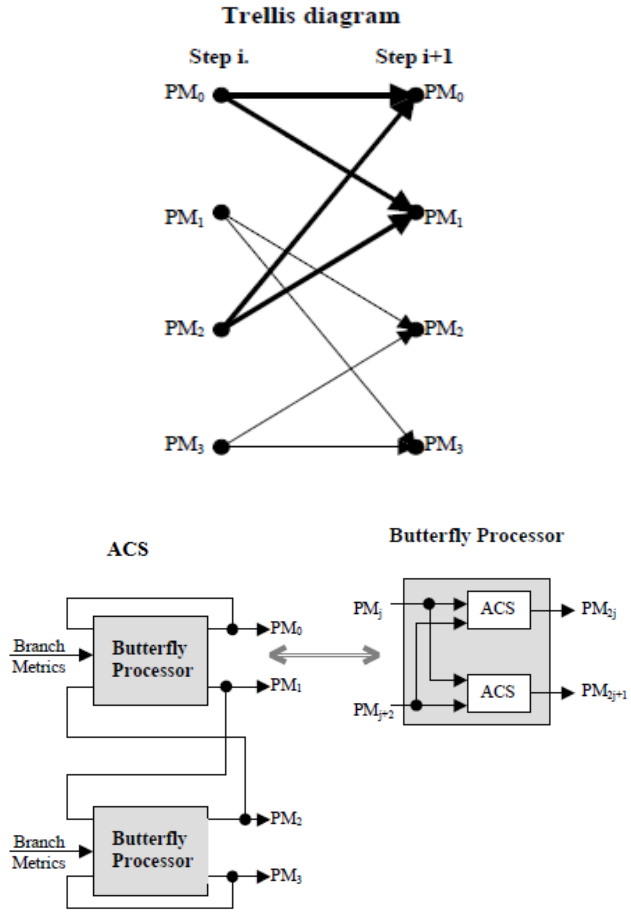


Figure 4: The link between a stage in the trellis diagram and the ACS unit

C. Traceback Algorithm

Traceback memory stores the history of decision bits from ACS operation. This memory is a two dimensional circular buffer, with rows and columns. The number of rows is equal to the number of states N = 2 K-1 where K is the constraint length. Each column stores the results of N comparisons corresponding to incoming coded bits at each time interval. The number of columns is equal to the "Trace back depth (L)" defined below. Using the trace back operation, every state from a current time is followed backwards through its maximum likelihood path. All of the paths converge at a point somewhere previous point in time. The point at which the corrected bit streams starts is called the merger point (also called the trace back depth). By finding this point, the trace back operation decisively determines the state of the encoder at a given time, by showing that at this point there is no choice for an encoder state given the global maximum likelihood path. The performance of Viterbi decoder largely depends upon the trace back depth. Normally for decoders using non punctured codes, the trace back depth equals 5-times constraint length, which is sufficient to decode the correct output in the presence of noise. But for decoders

using punctured codes and a 2/3 data rate, 8-times constraint length is optimal. If the data rate is 3/4, 10- times constraint length is recommended. Going beyond a 10-level constraint length does not make sense because performance gains are minimal. Thus, it is best to cap the constraint length at 10 In conventional approach, three types of operations are performed inside a TB decoder:

- **Traceback read (TB)**
  This is one of the two read operations and consists of reading a bit and interpreting this bit in conjunction with the present state number as a pointer that indicates the previous state number (i.e. state number of the predecessor). Pointer values from this operation are not output as decoded values; instead they are used to ensure that all paths have converged with some high probability, so that actual decoding may take place. The traceback operation is usually run to a predetermined depth, L, before being used to initiate decode read operation.
- **Decode read (DC)**
  This operation proceeds in exactly the same fashion as the traceback operation, but operates on older data, with the state number of the first decode read in a memory bank being determined by the previously completed traceback. Pointer values from this operation are the decoded values and are sent to the bit-order re-versing circuit. A decode read can serve as a dual decode and traceback read, this allows us to decode read multiple columns using one traceback read operation of L columns.
- **Writing new data (WR)**
  The decisions made by the ACS are written into locations corresponding to the states. The write pointer advances forward as ACS operations move from one stage to the next in the trellis, and data are written to locations just freed by the decode read operation. In conventional approach, to implement Traceback, needs circular 4 memory banks, each of length L (traceback depth) as shown in Figure. 5. Total latency for first decoded bit is 4L. Firstly, WR operation is performed on M0, TB is then performed on M0 meanwhile WR is performed on Ml, WR is then performed on M2 & TB is performed on Ml, WR is performed on M3 & TB is performed on M2 & DC is performed on M0. DC lags WR exactly three memory banks. In this way, three operations are preformed parallel in circular manners as shown in Figure 5.

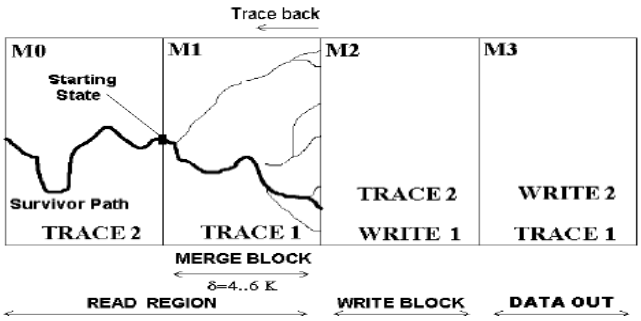


Figure 5: Memory organization for long trace back operations

4. Experimental Results

The convolution encoder and Viterbi decoder was implemented using the Spartan 2 FPGA series. The experimental results are shown in the Figure. 6. The different modules are designed using Verilog HDL simulated using ModelSim. The design implementation is done on Xilinx Spartan 2 xc2s200-5pq208. The different modules in the Viterbi decoder design are:

- Convolutional Encoder
- Branch Metric Unit(BMU)
- Add Compare Select Unit(ACSU)
- Butterfly unit
- Decoder unit

4.1 Simulation Results of Convolutional Encoder

Convolutional Encoder is first module in the design. The inputs of the module are *reset, clk* and *din*. The output generated is *out* which is of three bits. *Reset* is asynchronous input and *clk* is the encoder clock. *din* is the data to be given. Figure 6 shows RTL and simulation of Convolutional Encoder.

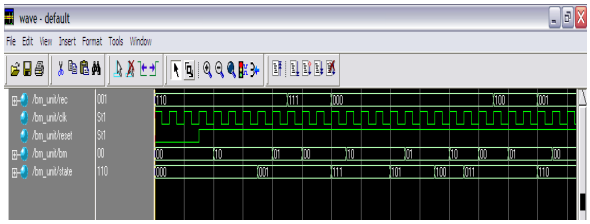
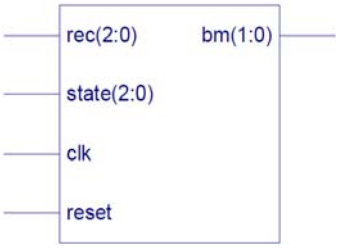


Figure 6: RTL and simulation of Convolutional Encoder

4.2 Simulation Results of BMU

Branch Metric Unit (BMU) is first module in the decoder. The inputs of the module are *rec, clk state* and *reset*. The output generated is *bm* which is of two bits. *reset* is asynchronous input and *clk* is the decoder clock. *rec* is three bit input which is fed from encoder output after inducing noise. Figure 7 shows RTL and simulation of BMU.



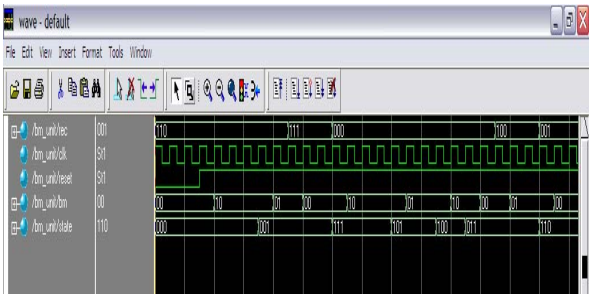
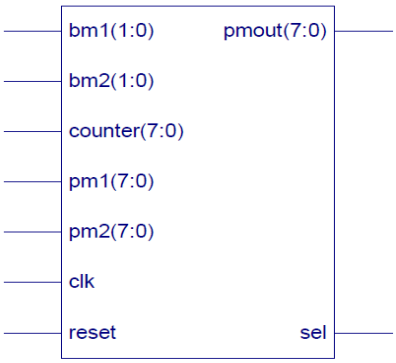


Figure 7: RTL and simulation of BMU

4.3 Simulation Results of ACSU

Add compare Select Unit (ACSU) is second module in the decoder. The inputs of the module are *bm1, bm2, counter, pm1, pm2, clk* and *reset*. The outputs generated are *pmout* which is of eight bits and *sel* which is the survivor path. *bm1 and bm2* are branch metrics from corresponding state which are of three bit each. These are fed from BMU output. *pm1* and *pm2* are previously accumulated path metrics which are of eight bits. Figure 8 shows RTL and simulation of ACSU.



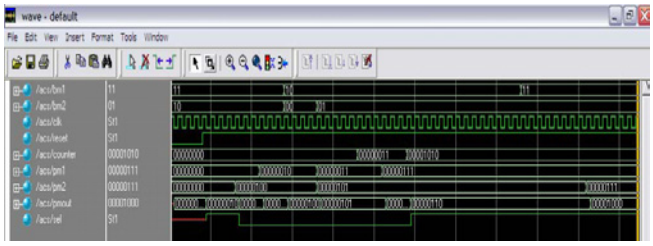
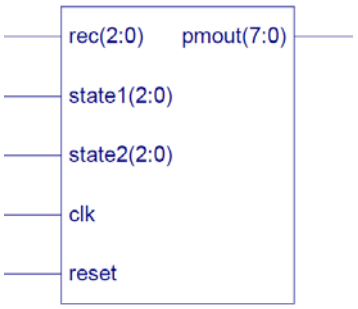


Figure 8: RTL and simulation of ACSU

4.4 Simulation Results of Butterfly Unit

Butterfly unit is the integrator of BMU and ACSU modules. The inputs of the module are *rec, state1, state2, clk* and *reset*. The output generated is *pmout* which is of eight bits. Inputs rec, state1 and state2 are of three bits each. Figure 9 shows RTL and simulation of Butterfly unit.



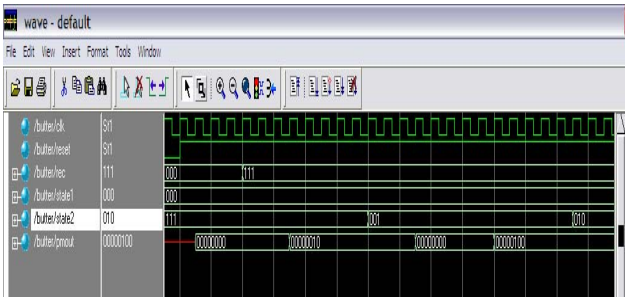
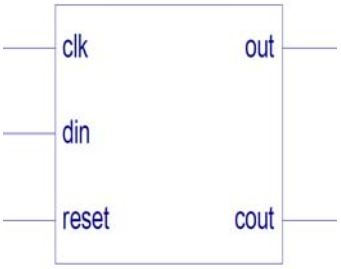
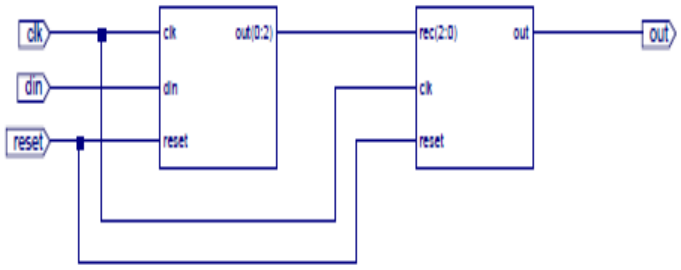


Figure 9: RTL and simulation of Butterfly unit.

4.5 Simulation Results of Top Module

Top module is the integrator of Encoder and Decoder module, including Noise module. The inputs of the module are *din, clk* and *reset*. The outputs generated are *out and cout*. *cout* is output of clock divider module which is used for implementing design in FPGA. Figure 10 shows RTL and simulation of Top module. The figure also shows the integration between Encoder and Decoder modules.





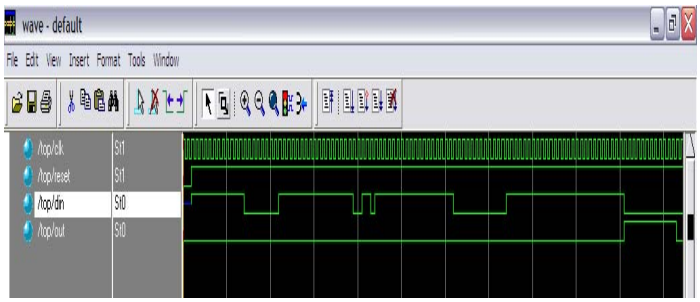


Figure 10: RTL and simulation of overall design.

5. Conclusion

Convolutional coding is a coding scheme often employed in deep space communications and recently in digital wireless communications. Viterbi decoding is the best technique for decoding the convolutional codes. Viterbi decoder with constraint length 3 and code rate 1/3 has been developed. The main consideration is to increase the speed.

The clk is the clock signal of whole system, and the reset is the reset signal of whole system. When the reset is set high, the convolution encoder and Viterbi decoder begin to work. The din and out respectively stands for input signals to convolution encoder and output Signals from Viterbi decoder. From the experimental results, it is clear that the input signal to the convolution encoder is identical to the output signal from the Viterbi decode. So this parallel design of TB and decoder module meets requirements.

The design is simulated using ModelSim. The design implementation is done on Xilinx Spartan 2 xc2s200-5pq208. The future work is focused on design a variable convolution encoder and Viterbi decoder. Because a variable convolution encoder and Viterbi decoder has a better flexibility in application.

References

[1] Lie_ou Wang, Zhe-ying Li, "Design and Implementation of a Parallel processing Viterbi decoder Using FPGA"IEEE Trans, Apr 2010.pp.77-80
[2] Hema S., Suresh Babu V and Ramesh P., "FPGA implementation of Viterbi Decoder", Proceedings of the 6th WSEAS int. Conf. on Electronics, Hardware, Wireless and Optical Communications, Corfu Island, Greece, Feb 2007.pp.162-167
[3] Ranjan Bose, "An Efficient Method to Calculate the Free Distance of Convolutional Codes", Proceedings of the 6th WSEAS int. Conf. on Electronics, Hardware, Wireless and Optical Communications, Corfu Island, Greece, Feb 2007.pp.60-63
[4] Bogdan, M. Munteanu, P. A. Ivey, N. L. Seed, N. Powell, "Power Reduction Techniques for a Viterbi Decoder Implementation", Electronic Systems Group, University of Sheffield, Mappin Street, Sheffield S1 3EA, UK.
[5] Samirkumar Ranpara and Dong Sam Ha, "A Low-Power Viterbi Decoder Design for Wireless Communications Applications", Int. ASIC conference, Sept. 1999, Washington, D.C
[6] Vasily P. Pribylov, Alexander I. Plyasunov (2005). "A Convolutional Code Decoder Design Using Viterbi Algorithm with Register Exchange History Unit". SIBCON, IEEE.
[7] S. K. Hasnain, Azam Beg and S. M. Ghazanfar Monir (2004). "Performance Analysis of Viterbi Decoder Using DSP Technique". INMIC. IEEE. pp 201-206.
[8] W. Zeng and J. Moon, "Modified Viterbi algorithm for jitter-dominated 1 channel," IEEE Trans. Magn., vol. 28, pp. 2895–2897, Sept. 1992.
[9] L. C. Barbosa, "Toward the detection of signals in presence of signal-dependent noise," in Proc. SPIE: Coding and Signal Processing for Information Storage, vol. 2605, Philadelphia, PA, Oct. 1995, pp. 48–56.
[10] P. R. Chevillat, E. Eleftheriou, and D. Maiwald, "Noise predictive partial response equalizers and applications," in IEEE ICC'92 Conf. Rec., June 1992, pp. 942–947.
[11] Eleftheriou and W. Hirt, "Noise-predictive maximum-likelihood (NPML) detection for the magnetic recording channel," in IEEE ICC'96 Conf. Rec., Dallas, TX, June 1996, pp. 556–560.
[12] J. D. Coker, E. Eleftheriou, R. L. Galbraith, andW. Hirt, "Noise-predictive maximum likelihood NPML detection," IEEE Trans. Magn., vol. 34, pp. 110–117, Jan. 1998.
[13] S. A. Altekar and J. K. Wolf, "Improvements in detectors based upon colored noise," IEEE Trans. Magn., vol. 34, pp. 94–97, Jan. 1998.

Author Profile

Mr. Vinay BK is a student in the Department of Electronics and Communication Engineering, School of Engineering and Technology, Jain University, Karnataka, India. He received his Bachelor degree in Electronics & Communication Engineering from VTU in 2012. He is pursuing M.Tech (SP and VLSI Design) in Electronics and Communication Engineering, Jain University, Karnataka, India. My research interest includes Low power VLSI Design; Analog and Mixed signal VLSI Design, Circuit design and simulations, DSP, and Embedded Systems Design. He's currently pursuing his project work at Texas Instruments.

Mr. Sunil MP, currently working as an Assistant Professor in the department of Electronics & Communication Engineering, School of Engineering and Technology, Jain University, Karnataka, India. He has received B.E degree in Electronics & Communication from VTU in 2009. He has received M.Tech degree in Electronics Design and Technology from National Institute of Technology, Calicut, and Kerala in 2011. His research interests include Embedded Systems Design, Analog and Mixed signal VLSI Design, Ultra- Thin Gate insulators for VLSI Technologies, RF VLSI Design, Microelectronics System Packaging, Microelectronics, Micro/Nano Sensor Technology, High-speed CMOS analog/RF-wave integrated circuits and systems.