

Hardware Implementation of High-Speed Low-Power Viterbi Decoder for Deep-Space Communication

Ali Ghasemi Khah¹, Yosef Seifi Kavian²

1- Shahid Chamran University, Ahvaz, Iran

Email: ali_ghasemikhah@yahoo.com

2- Shahid Chamran University, Ahvaz, Iran

Email: y.s.kavian@scu.ac.ir

Received: May 2016

Revised: October 2016

Accepted: November 2016

ABSTRACT:

In communication systems, ensuring correct information reception is very important, Error Correction Coding methods have been developed in order to achieve this goal. Convolutional Code is used in Wireless, Satellite, mobile phones and Deep-Space communications, it is one of the most powerful Error Correction code, and the Viterbi algorithm is robust way to decode it. Power consumption and speed are two important feature of Viterbi decoders, in many communications the power consumption is most important. In this paper, by removing extra decoding cycles, the SMU registers are reduced by 20%, the power consumption reduced by 14.5% and the speed increased 6 times without error correction performance loss. The proposed design is described by VHDL and it is implemented on Xilinx Spartan3, Xc3s400 FPGA chip.

KEYWORDS: Convolutional code, Viterbi Decoder, FPGA, Deep-Space Communication.

1. INTRODUCTION

In many communication systems, such as mobile phones and satellite transmission systems, different algorithms are used to correct errors of received data, since error correction coding better utilizes the band-limited channel capacity than special modulation schemes [1], [2].

Convolutional Codes are powerful and high performance in many communications such as Wireless, Satellite and Deep-Space communications, Viterbi Decoder is one of the best decoders for these codes and it decodes according to Viterbi algorithm. In deep space communications the power of the transmitted signal is the most severe limitation while there is no bandwidth restriction and therefore complicated coding schemes can be used to provide large error correcting capabilities. Convolutional codes have been used in many NASA missions such as Galileo and Voyager.

The special advantages of the Viterbi decoder are simplicity of implementation and its fixed decoding time [3-5]. Despite the simplicity, Viterbi Decoders has high chip area and power consumption. Viterbi Decoder complexity is increases exponentially by constraint length [5]. Viterbi Decoder strength in error correction has made designers to find low power designs [6-12].

Convolutional Encoder includes only several Flip-

Flops and XORs gates. The number of Flip-Flops represents the encoder constraint length. The constraint length equals to $m+1$ that m is number of Flip-Flops. The Code Rate is the ratio of encoder input to encoder output ($r=k/n$).

In convolutional encoder, trellis diagram is used for indicating input sequence and the corresponding outputs. Fig. 1 shows a trellis diagram for an input frame, this frame has 5 bits. Blue vectors indicate 1 Inputs and black vectors indicate 0 Inputs. Trellis diagram starts at S_{00} and it is arrived to S_{00} at end of each frame by reset bits. The path that is shown in Fig. 1 is Encoder path. Each trellis node has two inputs: upper input and lower input, Encoder path can move just on lower input at m th first states that m is number of Flip-Flops.

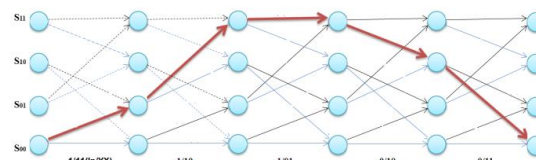


Fig. 1. Encoder trellis diagram and Encoder Path

One of the good codes for wireless communications that was used in Voyager mission has constrain length 7 and code rate $\frac{1}{2}$ with connection polynomials $g(171, 133)_8$ [3], This paper focuses on

the implementation of a Low-Power Fast Viterbi decoder with constraint length $K = 7$ and code rate $\frac{1}{2}$ for decode this code.

2. VITERBI ALGORITHM

The Viterbi algorithm was proposed by Andrew Viterbi in 1967 as a decoding algorithm for convolutional codes. Viterbi algorithm discovers data inputs from Encoder Path in trellis diagram. This path has a smaller path metric (PM) that calculates from Branch Metrics (BM). BMs are Hamming Distance between decoder inputs and calculated inputs. Fig. 2 shows calculated inputs on trellis branches and decoder inputs under the trellis diagram for each state. Also Hamming Distances are shown in Parenthesis next to calculated branches. The BMs accumulated in each stage to calculate PMs are shown in trellis nodes (circles) in Fig. 2.

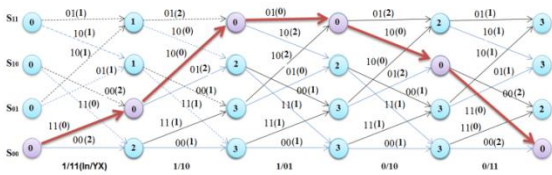


Fig. 2. Convolutional Encoder trellis diagram

The Viterbi algorithm reduces the complexity by eliminating the least likely path at each transmission stage so in each node smaller PM is selected, these PMs are known as survivor and others are non-survivor. If in a node, paths have same PMs, the survivor path will select randomly.

Viterbi algorithm starts to discover data inputs when all decoder inputs of a frame have received. Discovering starts at last bit of a frame backward to first bit. In each stage data input discovered from survivor paths. This method is known as Maximum Likelihood (ML).

3. VITERBI DECODER

Viterbi decoder decodes convolutional code according to found encoder path by maximum likelihood method. In this method Viterbi decoder estimates cost of all available paths and chooses lower cost path to the correct path, the correct path is Encoder path. The block diagram of the Viterbi decoder is shown in Fig. 3.

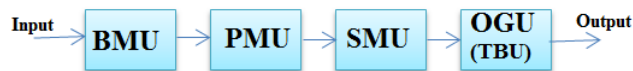


Fig. 3. Viterbi Decoder block diagram

3.1. Branch Metric Unit (BMU)

Transmitted data is became noisy from channel, so that received data to decoder is more like analog signal and

cannot expect a digital signal that has two levels: one and zero. Received signal can be quantized into two levels, either one or zero that is called hard decision. If the input signal is quantized and processed for more than two levels, it is called soft decision. In this paper ML algorithm with soft decision has been proposed so the decoder receives 2-bit input in each time. For decoding, the Hamming distance from the expected value of these inputs is needed that BMU do it. My decoder, at any time has two bits input so hamming distance from 00, 01, 10 and 11 is needed. The block diagram of BMU is shown in Fig. 4.

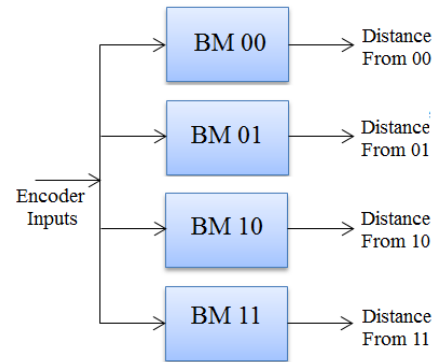


Fig. 4. BMU block diagram

3.2. Path Metric Unit (PMU)

According to Viterbi Algorithm, Encoder Path has lower cost or smaller PM. Cost of each path is calculated from sum of its branches. For the two inputs entering the same node, the input with the lower cost path survives, and the other one is discarded [13]. Decoder uses ACS block for sum branches and selects smaller PM. Architecture of an ACS block is shown in Fig. 5. Block diagram of PMU is shown in Fig. 6.

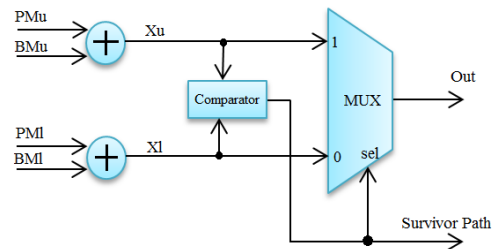


Fig. 5. Architecture of an ACS

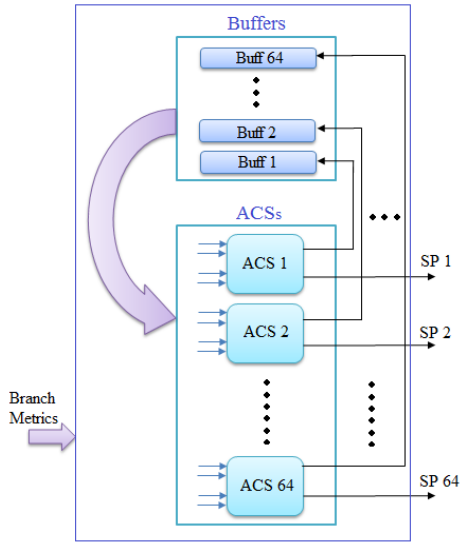


Fig. 6. PMU block diagram

3.3. Survivor path Memory Unit (SMU)

In each stage (cycle), ACSs output that indicate smaller PM should be saved, they are saved in Survivor Memory. This memory is included L 64-bit Registers that L is number bit of each frame. For example in a 30-bit frame, Survivor Memory includes 30 64-bit Registers. In this paper we consider the number of bits to be 30 per each frame. SMU block diagram is shown in Fig. 7.

Just one of the PMU Registers can be updated at any clock pulse period, this can be realized by the clock gating-scheme (shown in Fig. 7). This scheme will reduce power consumption [6], [7], [9], [14].

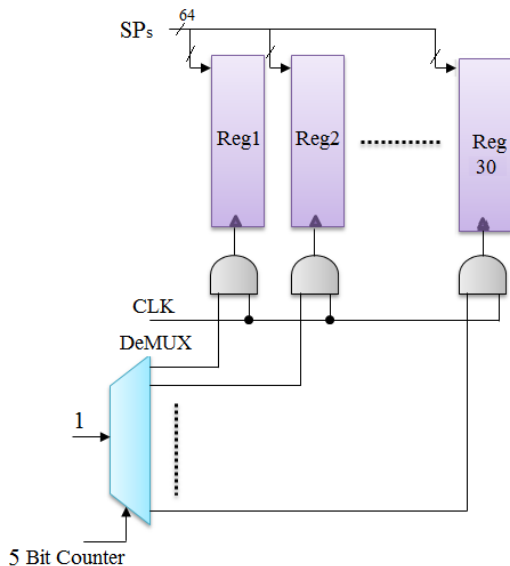


Fig. 7. Block diagram of SMU with Clock Gating scheme

3.4. Output Generation Unit (OGA)

There are two methods for output generation from SMU data: Trace Back (TB) and Register Exchange (RE). In this paper, TB method is used because it is a low power method.

TBU calculate the Survivor Path from SMU data. Trace back is done during a clock cycle, after receiving one frame and before arriving next frame. At the last stage of the trellis diagram (Fig. 1), the TB method estimates the decoded bits, starting from the state with the minimum PM, S00 [13]. Block diagram of TBU is shown in Fig. 8. TBU includes L combinational block, each block calculate corresponding output.

TBU is used just at one clock cycle and it is unusable at other 29 clock cycles, so it can be disabled to reduce power consumption. This scheme is Toggle filtering [15].

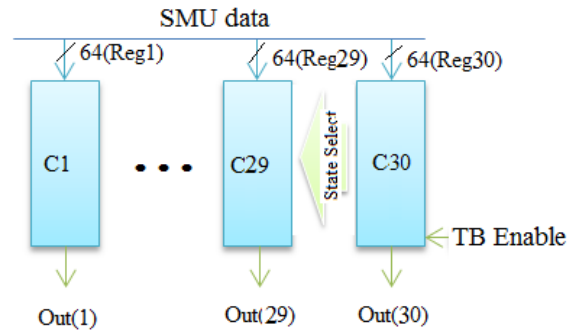


Fig. 8. TBU block diagram

4. PROPOSED VITERBI DECODER

4.1. Low Power SMU

In decoding process, ACSs cannot choose upper path at first 6 stages, so its SP outputs become 0. Therefore we do not consider memory for these stages, so our SMU includes 24 64-bit Registers instead of 30 64-bit Registers, that this reduces SMU by 20% which followed by lower power consumption.

4.2. High Speed Decoder

Maximum clock frequency is eliminated with TBU propagation delay, because propagation delay for TBU is very more than other units. Propagation delays are calculated 24.3nS and 4nS for TBU and PMU respectively. Trace back is done during first clock cycle of each frame, so Maximum clock frequency estimate from (1).

$$f_{clk_Max} = \frac{1}{T_p} = 41.1\text{MHz} \tag{1}$$

That T_p is propagation delay of TBU. As mentioned before, we do not update SMU registers at the first 6 clock cycles, therefore Trace back can do at these clock cycles, so the maximum clock frequency estimate from (2). Therefore maximum clock frequency is improved

by 6 times.

$$\frac{T_p}{6} = \frac{1}{f_{clk_Max}} \Rightarrow f_{clk_Max} = \frac{6}{T_p} = 247 \text{ MHz} \quad (2)$$

5. SIMULATION RESULTS

A MATLAB code is driven to evaluate the performance for Viterbi decoders. BER curve is estimated vs. SNR using AWGN channel and BPSK modulation for the convolutional code with both normal and proposed Viterbi Decoder. BER curve shows they have same performance (red and green curves in Fig. 9), this simulation is done for 100000 random frames.

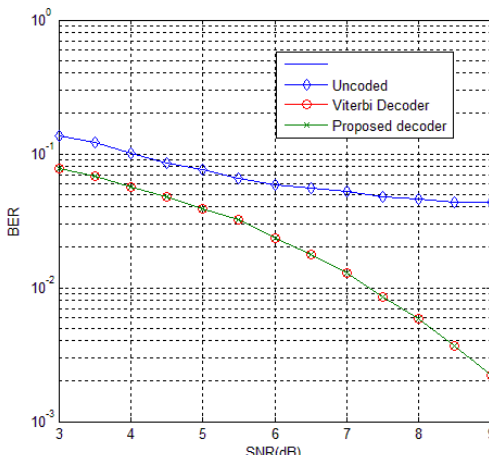


Fig. 9. MATLAB simulation for proposed viterbi decoder

In order to have a built-in self-test design, a test bench is used. The test bench block diagram is shown in Fig. 10. This test bench provides decoder inputs. The Serial Input Generation block generates the random inputs for encoder. The Serial Input Generation block contains a LFSR (Linear Feedback Shift Register) that is shown in Fig. 11[16].

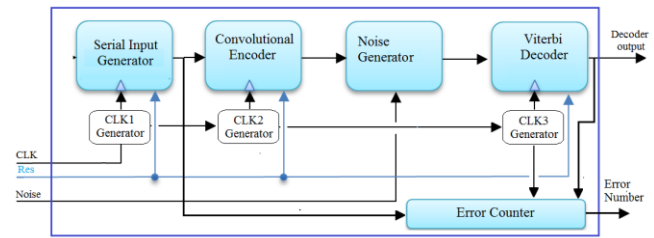


Fig. 10. Test Bench for Simulation

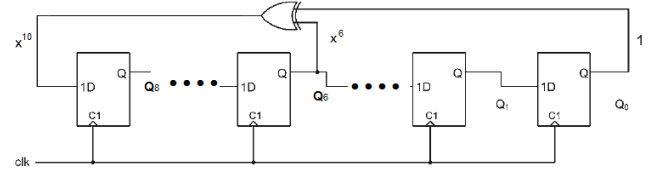


Fig. 11. Serial Input Generation block structure

Proposed Decoder and normal decoder are described with VHDL for estimation and comparison their power consumption. The simulation of the proposed Viterbi decoder has been made using Isim12.1 digital simulator to test the function of the implemented decoder. **Fig. 12** shows the simulation waveforms with applying an error pattern of 9 bits in error along the received frame. The result shows the efficient performance of the decoder in correcting up to 9 errors.

Xilinx XPower Analyzer12.1 has been used for power simulation. Table 1 shows the power simulation results at clock frequency 60MHz and Vccint =1.2v. The results in table1 show that the dynamic power in Low-Power design is reduced by 12%.

Table 1 : Power simulation results

Power consumption	Dynamic power (mW)	Static Power (mW)
Decoder		
Low power Decoder in [6]	34	60
Proposed Decoder	30	60

6. EXPERIMENTAL RESULTS

Xilinx ISE12.1 has been used for synthesization standard and proposed Viterbi Decoder. Proposed design is implemented on a Xilinx Spartan3, Xc3s400 FPGA. FPGA kit is shown in Fig. 13. Experimental measured dynamic power for normal and proposed

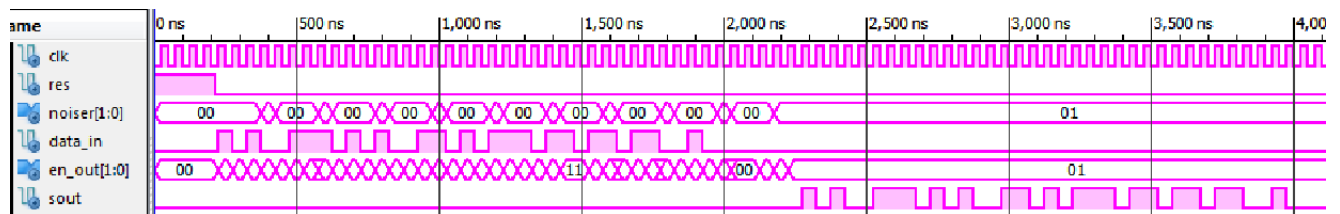


Fig. 12. Decoder output with applying an error pattern, along the received a frame

design is shown in table2. The experimental results show that the dynamic power in Low-Power design is reduced by 14.5%. The experimental results confirm the simulation results approximately.

Table 2 : Experimental power measured results

Decoder	Power consumption	Dynamic power (mW)
Low power Decoder in [6]		345
Proposed Decoder		295

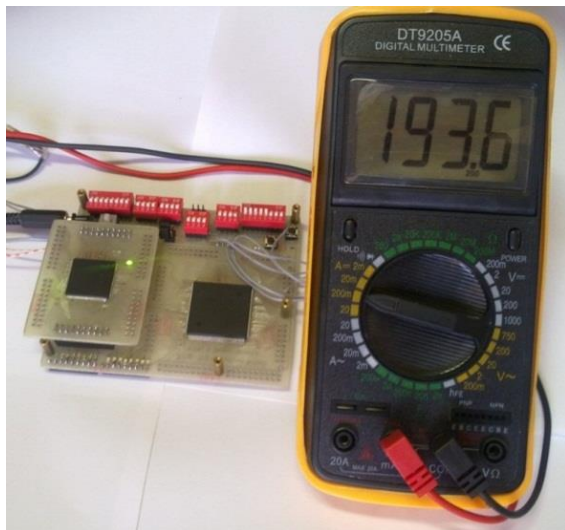


Fig. 13. Spartan3 FPGA kit

REFERENCES

- [1] U. Meyer-Baese, "Digital Signal Processing with Field Programmable Gate arrays ", *Third Edition* , Springer, pp. 418-436, 2007
- [2] H. Luke, "Signalübertragung", Springer, Heidelberg, 1988.
- [3] P. Sweeney, "Error Control Coding", Wiley, 2002.
- [4] R. H. Morelos-Zaragoza, "The Art of Error Correcting Coding", Wiley, 2002.
- [5] S. Lin, D. J. Costello, "Error Control Coding", Prentice-Hall, pp. 315-348, 1983.
- [6] S. W. Shaker, S. H. Elramly, K. A. Shehata, "FPGA Implementation of a Reconfigurable Viterbi Decoder for WiMAX Receiver", *IEEE 21st International Conference on Microelectronics, ICM 2009*. Marrakech, Morocco, 19-22 December 2009.
- [7] S. Ranpara, D. S. Ha, "A Low-Power Viterbi Decoder Design for Wireless Communications Applications", *Int. ASIC conference*, pp. 377-381, Washington, D.C, Sept. 1999.
- [8] M. Guo, M. O. Ahmad, M. N. S. Swamy and C.Wang, "FPGA Design and Implementation of a Low-Power Systolic Array-Based Adaptive Viterbi Decoder", *IEEE Transactions on circuit and system regular papers*, Vol. 52, NO. 2, pp. 350-365, 2005.
- [9] S. Ranpara, "On a Viterbi decoder design for low power dissipation," *M.Sc, Thesis*, Virginia Polytechnic Institute and State University, 1999.
- [10] F. Sun, T. Zhang, "Low-power State Parallel Relaxed Adaptive Viterbi Decoder," *IEEE Trans. Circuits and Syst. I*, Vol. 54, No. 5, pp. 1060-1068, May 2007
- [11] M. D. Shieh, T. P. Wang, and D. W. Yang, "Low-power register-exchange survivor memory architectures for Viterbi decoders," *IET Circuit, Devices Syst.*, Vol. 3, No. 2, pp. 83-90, April 2009
- [12] C. C. Lin, Y. H. Shih, H. C. Chang, and C. Y. Lee, "Design of a Power-Reduction Viterbi Decoder for WLAN Applications," *IEEE Trans. Circuits and Syst. I*, Vol. 52, No. 6, pp. 1148-1156, June 2005
- [13] B. Singh, S. Agarwal and T. Varma, "Hardware Implementation of Viterbi Decoder for Wireless Applications", *International Journal of Computer Communication and Information System (IJCCIS)*, Vol. 2. No. 1, ISSN: 0976-1349 July – Dec 2010
- [14] J.Oh and M.Pedram., "Gated clock routing for low-power microprocessor design", *IEEE Transactions on Computer-Aided Design*, pp. 715-722, 2001.
- [15] F. Ghanipur, A. R. Nabavi, "Design of a Low-Power Viterbi decoder for wireless communication," *Electronics, circuits and systems*, Dec 2003, pp 304-307, Vol. 1, doi: 0-7803-8163-7/03/\$17.00 © 2003 IEEE.
- [16] Z. Navabi, "Digital System Test and Testable Design", Springer, 2011.