



# OPTIMIZATION OF VITERBI DECODER WITH FAST RADIX 2 ACS UNIT

N. D. Bobby, N. Prabhakaran E. Praveen Kumar and S. Thiyagarajan

Department of ECE, Vel Tech High Tech Dr Rangarajan, Dr Sakunthala Engineering College, Chennai, India

E-Mail: [ndbobby@gmail.com](mailto:ndbobby@gmail.com)

## ABSTRACT

Viterbi decoder is one of the unique techniques used to retrieve the data in a redundant way by using error detecting and correcting codes. It is used for decoding codes of long and short constraint length. To minimize power consumption and BER, we have implemented Fast radix 2 add compare select unit in Viterbi decoder. The resource utility and computational speed was less in Viterbi decoder with fast radix 2 ACS unit. The power utilization reduced drastically to 50%, was observed where as in Viterbi decoder with conventional ACS unit the power consumed was 80%.

**Keywords:** VLSI, viterbi decoding, FPGA, VD, ACS, Fast radix 2 ACS.

## 1. INTRODUCTION

Viterbi decoding is used for decoding convolutional code in maximal likelihood sense. The decoding complexity of the viterbi decoder increases with respect to the code length. In low-power designs, good power reduction methods are needed. In order to overcome this problem, modified add compare select unit has been developed. This Decoder with modified acs unit reduces the average number of computations. Decoded information has low bit error rates (BER) compared to Viterbi algorithm implementations. Power reduction in Viterbi decoder has been achieved by either reducing the number of states (reduced-state sequence decoder) [3], the size of survivor memory [4], or the number of trellis paths (limited-search trellis) [5] at the expense of increased BER. Other approaches include scarce state transition [6] where the most-likely path passes through the zero states most of the time allowing shorter survivor memories and efficient limited-search trellis decoding [7]. In this paper, architectures for the Fast radix2 add-compare-select unit and its implementation in Viterbi decoder are discussed.

## 2. VITERBI DECODER ARCHITECTURE

The view of the Viterbi decoder architecture is shown in Figure-1. Most hardware implementations of the Viterbi algorithm [9] are split into three parts: the branch metric generators (BMG), add-compare-select (ACS) units, path metric storage and control, and the survivor memory unit. BMG unit determines distances between received and expected symbols. ACS unit determines path costs and identifies lowest-cost paths. The survivor memory stores lowest cost bit-sequence paths based on decisions made by the ACS units.

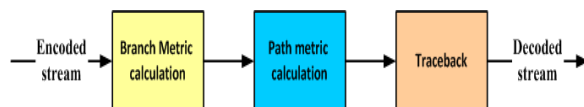


Figure-1. Viterbi decoder architecture.

In the implemented decoder, the expected symbol value  $bm$  select signal is used to select the appropriate branch metric from the BMG, This branch metric value is

combined with the path metric of its parent present state to form a new path metric,  $d_i$ .

At each Trellis stage, the minimum-value surviving path metric among all path metrics for the preceding Trellis stage,  $d_m$ , is computed. New path metrics are compared to the previous path  $d_m$  to identify path metrics with excessive cost. Comparators are then used to determine the life of each path.

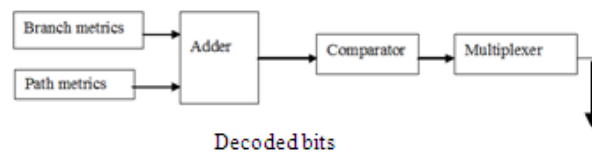


Figure-2. ACS unit of Viterbi decoder.

## 3. FAST RADIX 2 ACS UNIT

The fast radix 2 add compare select unit is shown in Figure-3. The two paths are  $i$  path and  $j$  path respectively. The least significant bit of both branch metric and path metric of  $i$  path are added in adder one respectively. The output of 5 bit with carry comes as adder one output.

Same procedure is repeated for  $j$  path in adder two which is in Figure-3. The same five bit output with carry is generated in adder two similar to adder one. The output of the adder one and adder 2 are given as input to the least significant bit comparator. The bit wise comparison is done. If any bit differs lsb output is zero, if all bit are equal lsb output is one.

The most significant bits of both  $i$  path and  $j$  path are given to the subtractor and the same to the equality comparator. The output of the equality comparator is taken as selection line for multiplexer one. The multiplexer one is 2:1 type.

The inputs for multiplexer one is borrow bit of the subtracted output and least significant bit comparator output. Selection line is zero, LSB comparator output is passed, if selection line is one, the borrow bit of the subtractor output comes as multiplexer one output. Multiplexer one output becomes the decision bit for multiplexer two.



Multiplexer two is 8:1 type. The inputs for multiplexer two is i path and j path. If decision bit is zero, then i path is selected. If decision bit is one, then j path is selected. This way best path is selected using fast radix 2 add compare select unit. Also, by looking at previous state metric values and comparing with present state metric value, error free path is chosen as best path with low cost.

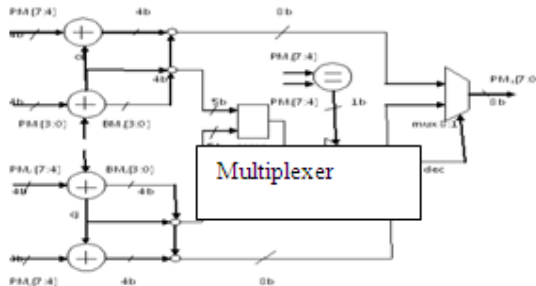


Figure-3. Fast radix 2 add compare select unit.

#### 4. IMPLEMENTATION OF FAST RADIX 2 ACS IN VITERBI DECODER

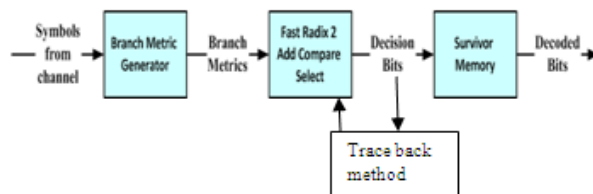


Figure-4. Viterbi decoder with fast radix 2 acs unit.

In Viterbi decoder, code words received from free space are given to branch metric generator to generate branch metric bits. Branch metric bits are nothing but the bits that are received and compared with the ideal pair of bits, which is Hamming distance for hard decision decoding methods. For every decoding state, surviving path with minimum path metric value is computed. Computations are performed by adding the branch metric bits with path metric bits which generates new path metrics. These path metrics are compared with earlier path metrics and low cost value is selected whereas high path metric value is discarded. In normal Viterbi decoder, we have conventional add compare select unit [7].

In this method, by replacing the add compare select unit with proposed design fast radix 2 add compare select unit and we observe the experimental results. Path metrics for all, potential next state paths are computed by fast radix 2 ACS unit. Comparators are then used to determine life of each path. Using the trace back concept, the decoded bits follows valid path; at end stage of computations, we get fewer paths with almost similar path cost. These paths are given as inputs to the fast radix 2 add compare select unit and we trace out the decoded bits with one valid path [8].

Encoded output bits are given as inputs to the Viterbi decoder with fast radix 2 add compare select unit.

Viterbi decoder processes these messages and output of decoder is obtained. Encoder input data and decoded output data both are one and the same. The path metric value is zero, the decoded bits follows the way of valid path ensuring error free transmission.

#### 5. IMPLEMENTATION ANALYSIS

Power consumption values for viterbi decoders were implemented using XC3S50. The implementation is done for Viterbi decoder with conventional acs and Viterbi decoder with fast radix 2 acs. In Table-1 we are comparing the both Viterbi decoder with Fast radix2 ACS and conventional acs. This shows the number of CLB slices, LUT, power consumption, number of filpflop and latency utilized for the above mentioned. Here we have analysed the process in xilinx project navigator XC3S50VQ100, the power analysis in Xilinx power and the simulation in modelsim software.

Table-1. Comparison of Viterbi decoder and fast radix 2 ACS in Viterbi decoder.

Analysis	Viterbi decoder	Fast radix 2 ACS in Viterbi decoder
No. of Slices	131	86
No. of Flip Flops	25	17
No. of Lut's	234	153
Power Consumption (mw)	56.5	23.5
Latency (ns)	11.99	2

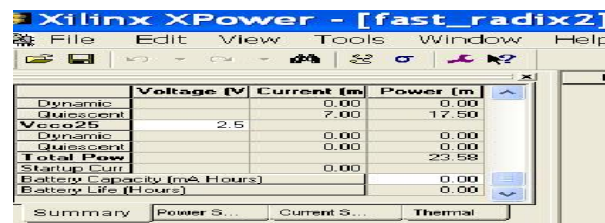


Figure-5. Power consumption of conventional acs in Viterbi decoder.

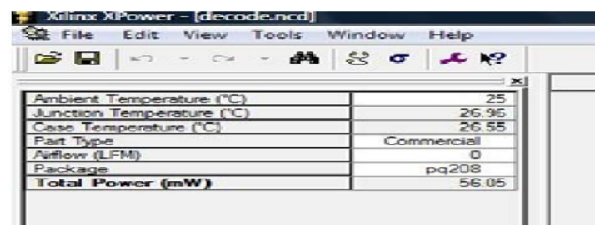
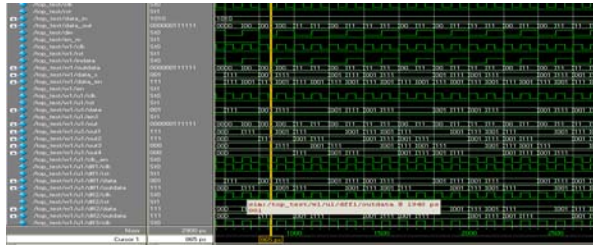
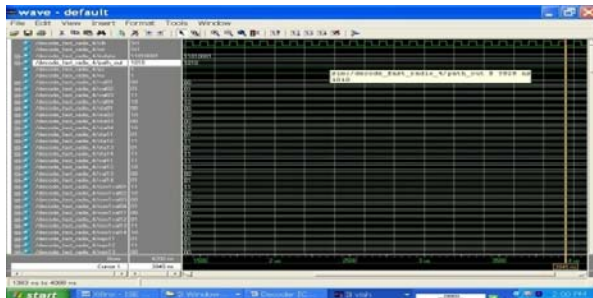


Figure-6. Power consumption of fast radix-2 ACS in Viterbi decoder.



**Figure-7.** Simulation of Viterbi decoder with conv. acs unit.



**Figure-8.** Simulation of decoded output of fast radix 2 add compare select unit in Viterbi decoder.

## 6. RESULT ANALYSIS

The simulation result for Viterbi decoder for both conventional acs unit and fast radix-2 add compare Select Unit is shown in Figures 7 and 8. The Power analysis is performed for conventional acs unit with Viterbi decoder and fast radix-2 add compare select unit in Viterbi decoder is shown in Figures 5 and 6. Both the decoders are synthesized and resource used is mentioned in the Table-1. The comparison shows that implementing fast radix-2 acs in Viterbi decoder has reduced the computational resources, memory, noise tolerance and power consumption in Viterbi Decoder.

## 7. CONCLUSIONS

The power reduction for fast radix2 add compare select unit in Viterbi decoder is observed to be minimized by 50%. From the analysis resource utilization is less for fast radix2 acs when implemented in Viterbi decoder and more area occupied for normal Viterbi decoder. Streamlining to find the minimum path and store the best path in survivor memory occupies more space area. In fast radix-2 acs unit in Viterbi decoder the power drastically reduced, occupied slices and flip flop utility are also found to be less. So we conclude Fast radix2 acs unit is best for decoding and to achieve minimum power consumption

## 8. FUTURE WORK

In future we plan to implement this in ARM processor and expecting the less power consumption. The application area will be wireless communications, Industrial local area networks and soft ware designed radio.

## REFERENCES

- [1] Russell Tessier, Sriram Swaminathan, Ramaswamy, Dennis Goeckel and Wayne Burleson. 2005. A Reconfigurable, Power-Efficient Adaptive Viterbi Decoder. IEEE Transactions on VLSI Systems. pp. 484-488.
- [2] Rami A. Abdallah and Naresh R. Shanbhag. 2009. Error-Resilient Low-Power Viterbi Decoder Architecture. IEEE Transactions on signal processing. 57(12): 4906-4917.
- [3] Irfan Habib and Ozgun Paker. 2010. Design space Exploration of Hard Decision Viterbi Decoding: Algorithm and VLSI Implementation. IEEE Transactions on VLSI. 18(5): 794-807.
- [4] Samir Kumar Ranpara. 1999. On a Viterbi decoder design for low power dissipation. Synopsis. pp. 1-25.
- [5] F. Chan and D. Haccoun. 1997. Adaptive Viterbi decoding of convolutional Codes over memoryless channels. IEEE Transactions on Communications. 45(11): 1389-1400.
- [6] Sriram Swaminathan and Russel Tussier. 2002. A dynamically reconfigurable adaptive Viterbi decoder. ACM /SIGDA. pp. 227-236.
- [7] A. J. Viterbi. 1967. Error bounds for convolutional coding and an asymptotically optimum decoding algorithm. IEEE Trans. Information. Theory. IT-13: 260-269.
- [8] Joshi S.P. 2011. Low power Viterbi decoder by modified ACSU architecture and clock gating method. International conference on communication and signal processing ICCSP 2011. pp. 499-503.
- [9] S. Nanda, K. Balachandran and S Kumar. 2000. Adaptation techniques in Wireless packet data services. IEEE Communications Magazine. 38(1): 54-64.
- [10] D. Matolak and S. Wilson. 1996. Variable-complexity Trellis decoding of binary convolutional codes. IEEE Transactions on Communications. 44(2): 121-126.
- [11] Amithaba Bhattacharya. 2012. Digital communication. copyright@2006 by Tata McGraw Hill Education private limited, tenth reprint. pp. 220-236.



- [12] B. Pandita and S. K. Roy. 1999. Design and Implementation of a Viterbi Decoder Using FPGAs. Proceedings of IEEE International Conference on VLSI Design. pp. 611-614.
- [13] J. Proakis. 1995. Digital Communications, 4<sup>th</sup> edition, McGraw-Hill, New York. pp. 485-492.
- [14] A. J. Viterbi. Convolutional codes and their Performance rent sections.