

A Parallel Viterbi Decoder Implementation for High Throughput

Shoaib bin Altaf, Muhammad shoaibbinalt@wisc.edu

Motivation

Convolutional coding with Viterbi decoding is a popular and powerful method for Forward Error Correction (FEC) in communication systems. The decoding process of a convolutionally encoded message is based on the maximum likelihood estimation principle. This could be achieved by selecting the most likely path in the trellis structure that describes all possible states. However the implementation of the Viterbi decoder required management of the relevant memory contents which is a major design problem for both hardware & software realizations. Latest wireless communication technologies like WiMAX is opening up new challenges in baseband hardware design and demanding high speed, area efficient and reconfigurable designs. As the data speeds are moving skywards, demand for high speed Viterbi decoders is increasing. The proposed design particularly will strive for a design optimized for high speed such that the at each clock cycle the decoder will be decoding a couple of bits instead of a single bit, without compromising on performance so that it suits the requirements of latest wireless standards like 802.16e.

Prior Work

Researchers have been working in the area of optimizing the Viterbi decoder for couple of years. Some tried to increase the throughput & reduce the area consumed by giving the idea of radix-4 approach instead of radix-2 [4]. The idea of systolic array processing is also not new in this regard for increasing the throughput [5]. Nandu et al. employed normalization & systolic array processing in their design [2]. Batcha et al. achieved high throughputs compatible with Wimax using pipelining [3] & Fettweis [1] also used pipelining for increasing the throughput of their design. Altera & Xilinx have also provided their configurable IP cores for the Viterbi decoder [6] [7].

Approach

The 1st task will be to develop the complete understanding of the Viterbi decoding Algorithm. After studying the algorithm a Matlab/C code will be written/ used for the determination of clock cycles used in the computation & will be treated as a standard for the comparison. The next target will be to locate the portions of the code which can be the main target for parallelizing by applying the various optimization approaches we studied in the course like

- Look-ahead Transformation
- Loop transformation
- Pipelining
- Algorithm reformation

For this design I am only concentrating on the increasing the speed of the decoder irrespective of the area/memory utilization, for which Look-ahead Transformation will be the major technique. So for initial considerations, the design parameters such as constraint length, code rate, hardware utilization etc. will not be a concern at this point. Since the major objective is to look for the possibility for coming out with such an algorithm which instead of decoding bit by bit, decodes a number of bits.

If I am successful in coming up with a technique then it will be checked for its validity for a simple case & if unsuccessful then I will come up with some arguments that why it's not possible & under what conditions the look-ahead methodology can be exploited.

Expected Results

Since the output of the decoder will be vectorized instead of being a single output. I am expecting very high throughputs. The results will include a speed comparison of the studied techniques. A comparison of the BER for various SNR values will also be a part of the final report in order to establish that the adopted approach of increasing the throughput didn't compromise the performance of the decoder.

References

1. G. Fettweis and H. Meyr, "High-Speed Parallel Viterbi Decoding: Algorithm and VLSI-Architecture", IEEE Communication Magazine, May 1991
2. S.Nandula, Y.S. Rao, and S.P.Embanath, "High speed area efficient configurable Viterbi decoder for WiFi and WiMAX systems", June 2007
3. M.F.Batchia and A.Z.Sha'ameri, "Configurable Adaptive Viterbi Decoder for GPRS, EDGE and Wimax" IEEE Conf., May 2007
4. S.C.Kim, J.H.Ryu, and J.D.Cho, "Low Power, High-Rate Viterbi Decoder Employing the SST (Scarce State Transition) Scheme and Radix-4 Trellis"
5. C.Y.Chang and K.Yao, "Systolic Array Processing of the Viterbi Algorithm", IEEE Trans., 1989
6. http://www.altera.com/literature/ug/ug_viterbi-compiler.pdf
7. <http://www.xilinx.com/ipcenter/viterbi/features.htm>