ISSN 2321 – 2004
ISSN 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 3, June 2013

LOW POWER ADAPTIVE VITERBI DECODER DESIGN FOR TRELLIS CODED MODULATION

R.Mahendran[1], T.M.Sathish kumar[2]

Student, ECE (PG Scholar), K.S.R. College of Engineering, Tiruchengode[1]

Assistant Professor, ECE, K.S.R. College of Engineering, Tiruchengode[2]

Abstract: By using viterbi decoder the power can be reduced according to number of stages used in each block. The efficient way to reduce data corruption in digital communication channels using various decoding algorithm. Although hardware implementations of decoding algorithms, (i.e.,) Viterbi algorithm, have shown good noise tolerance for error-correcting codes, these hardware implementations require an exponential increase in VLSI area and power consumption to achieve increased decoding accuracy in different stages. To achieve reduced decoder power consumption, we noticed the hardware implementation decoders based on the reduced-complexity adaptive Viterbi algorithm (AVA). In proposed decoding method the overall power reduced in each module can be improved when compared to previous method.

Keywords: Trellis Coded Modulation (TCM), Adaptive Viterbi Decoder (AVD), VLSI.

I. INTRODUCTION

As the error-correcting capability of convolutional codes is improved by employing codes with larger constraint lengths K, the complexity of decoders [1] is increased. The Viterbi algorithm reduced the decoding speed and also overall area can be increased. To address this issue, the reduced-complexity adaptive Viterbi algorithm (AVA) has been developed. By using this decoding technique the overall power can be reduced and also number of computation should be reduced. The average number of computations per decoded bit for this algorithm is substantially reduced versus the Viterbi algorithm, while comparable bit-error rates (BER) are preserved.

Convolutional codes are frequently used to correct errors in noisy channels. They error correcting capability and perform well even on very bad channels (with error probabilities of about 10^{-3}). The convolutional codes are extensively used in satellite communications. An encoding process is a simple procedure, in output side decoding of a convolutional code is much more complex task.

The Viterbi decoding algorithm [1], proposed in 1967 by Viterbi, is a decoding process for Convolutional codes in memory-less noise. In this thesis, a low-power design of Adaptive Viterbi Decoders in the standard cell design environment is proposed.

The standard cell design environment shows the behavior of a design is described in a high-level hardware description language, such as VHDL or Verilog. The gate level design is synthesized to generate a gate level design. The gate-level design in viterbi decoder is placed and routed to generate a layout of the various design.

The advantages of a standard cell based design over full custom design are faster turnaround time for the design, ease in design verification and more accurate modeling of the circuit. We introduce low power design techniques into the behavior of Adaptive Viterbi decoder.

After the behavior of an Adaptive Viterbi decoder was described in VHDL, we modified the behavior of the circuit to reduce dynamic power dissipation. Adaptive Viterbi detectors are used in cellular telephones with low data rates, of the order below 1Mb/s but with very low energy dissipation requirement.

ISSN 2321 – 2004
ISSN 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 3, June 2013

II. BACKGROUND

In recent years, the addition of systematic redundancy to the transmitted information in the form of error-control coding has proved to be a potential way to overcome channel disturbances that produce errors, in digital communication systems. The evolution of error-control codes such as the convolutional codes, the number of stages present in the corresponding decoders has also been increasing. The implementation of such a decoder is to find a codeword that most closely resembles the received input sequence. The Viterbi algorithm (VA), which is one of the most extensively, employed decoding algorithms, along the decoding with short constraint length K. The memory requirement and number of computations poses a big obstacle when decoding codes with larger constraint lengths. To overcome this problem the concept of adaptive Viterbi decoding algorithm (AVA) was developed.

In TCM decoder architecture [2], TCM encoders usually employ high-rate convolutional codes that requires a many more transition paths per state than low-rate codes do. The algorithms are more feasible to purge unnecessary additions by applying the T algorithm on BMs. Choosing the T-algorithm on BMs instead of PMs allows one to move the "search-for-the-optimal" operation out of the add–compare–select-unit (ACSU) loop. Hence, the clock speed will not be affected.

The Viterbi algorithm is an decoding process for solving maximum likelihood sequence estimation (MLSE) problems. The encoder for a rate 1/2 code generates two output codeword bits as a function of the giving input information bits and the encoder state (stored bits in the registers).In a hard drive, the disk is spinning so fast that the binary values of the data tends to get blurry.

An **ML** path is found with the aid of a branch metric and a path metric. A branch metric is the Hamming distance between the estimate and the received code symbol. The branch metrics accumulated along a path form a path metric. A path metric at each state, often referred as state metric in path metric for the path from the initial state to the given state.

III. PROPOSED DESIGN

The design of an adaptive Viterbi decoder that uses survivor path with parameters for wireless communication in an attempt to reduce the power and cost and at the same time increase the decoding speed. This Algorithm is the optimum-decoding algorithm for convolutional codes and has often been served as a standard technique in digital communication systems for maximum likelihood sequence estimation. The Add-Compare-Select (ACS) and Trace Back (TB) units and its sub circuits of the decoder have been operated in deep pipelined manner to achieve high transmission rate.

The adaptive Viterbi algorithm was introduced with the goal of reducing the *average* computation and path storage required by the Viterbi algorithm. The number of stages that are computing and retaining all 2^{K-1} possible paths, it requires those paths which satisfy certain path cost conditions are retained at each stage, where a path's "cost" is defined as the Euclidean distance between the path and the received sequence.

The adaptive Viterbi decoder accepts two inputs from the channel which represent the outputs of the encoder that have been transmitted. To demonstrate the benefit of the adaptive Viterbi algorithm we have developed the first hardware implementation of the algorithm. This architecture takes advantage of parallelization and specialization of hardware for specific constraint lengths and dynamic reconfiguration to adapt decoder hardware to changing channel noise characteristics.

Although the adaptive Viterbi decoder can provide impressive decode rates with a static architecture implemented in reconfigurable hardware, improved performance can be achieved if the decoder is reconfigured to match required computation. In a second set of experiments, channel noise, as indicated by *SNR*, was used to indicate when the entire AVD architecture on the Wild One board should be reconfigured. Reconfigurable computing has been proposed for signal processing with various objectives which including high performance, more flexibility, specialization, and most adaptability.

ISSN 2321 – 2004
ISSN 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 3, June 2013

IV.ADAPTIVE VITERBI DECODER ARCHITECTURE

The use of error-correcting codes has proven to be an effective way to overcome data corruption in digital communication channels. In this manuscript, a power-efficient implementation of an adaptive Viterbi decoder has been described. To measure its power consumption, the AVA architecture has been implemented in two contemporary FPGA architectures for a range of constraint lengths.

To explore the power benefits of AVA use we have developed a hardware implementation of the algorithm. This architecture exhibits significant parallelism and supports dynamic reconfiguration to adapt decoder hardware to changing channel noise characteristics. Hardware reconfiguration provides the key mechanism to achieve decoder power savings.

The architecture of the implemented adaptive Viterbi decoder is shown in Figure

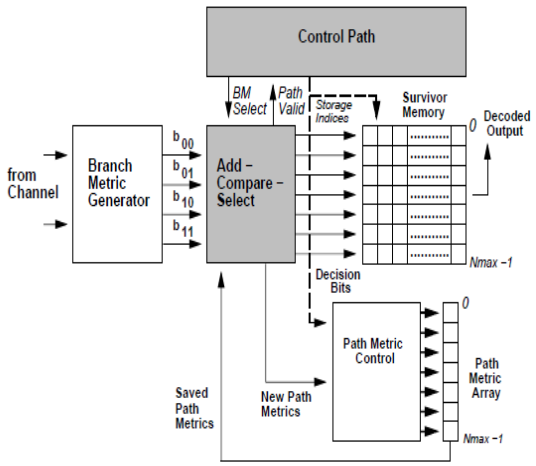

Figure 1 Adaptive Viterbi Decoder architecture

An Adaptive Viterbi decoder implementation consists of three basic building blocks:

1) Branch Metric Computation Unit (BMCU)

2) Add-Compare-Select Unit (ACSU)

3) Survivor Memory Unit (SMU)

These three units are very important process that is used to design a various systems to reduce the whole power in different algorithm.

A. *Branch Metric Computation Unit (BMCU)*

BMU is typically the smallest unit of Viterbi decoder. Its complexity increases exponentially with (reciprocal of the coding rate) and also with the number of samples processed by decoder per clock cycle (radix factor, e.g., radix-2 corresponds to one sample per clock cycle). The complexity increases linearly with *soft bits*. So, the area and throughput of the BMU can be completely described by these two factors.

As BMU is not the critical block in terms of area or throughput, its design looks quite straightforward. The version calculating the Hamming distance for coding rate ½ presented and it performs perfectly in terms of both area and throughput. A BMU calculating squared Euclidean or Manhattan distance is slightly more complex but can be easily mapped to an array of adders and subtracters as well.

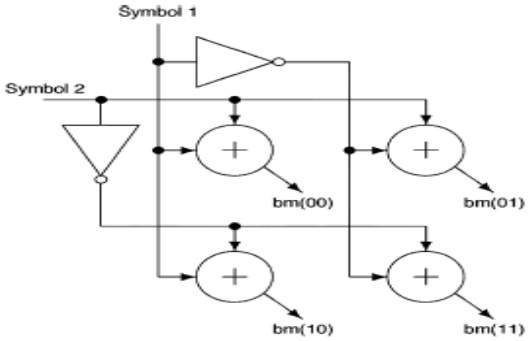

Figure 2 Block diagram of Branch Metric Unit

BM should be able to hold the maximum possible difference (distance) between ideal and received symbols. For hard input (Soft bits=1), the Hamming distance between a coded word and its complement is the maximum possible BM (i.e., the distance between code words consisting of all 0's and all 1's). In such case, the maximum distance is equal to the symbol length n.

B. Add-Compare-Select Unit (ACSU)

The add-compare-select unit is the heart of the Viterbi algorithm and calculates the state metrics. These state metrics of each state accumulate the minimum cost of 'arriving' in a specific state. The

ISSN 2321 – 2004
ISSN 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 3, June 2013

major task of the ACS is to calculate the metrics and selected paths. The add-compare-select (ACS) unit recursively accumulates the branch metrics to path metrics for all the incoming paths of each state and selects the path with minimum path metric as the survivor path.

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.

In order to reduce the switching activity of the ACS during path pruning, an additional signal S is used to indicate whether the path is pruned or not.S is determined by the sign (i.e., MSB) of the calculated path metric. S is also used to gate the path metric registers to reduce the switching activity of the ACS if the path is pruned.

Two distinctive features of our decoder are the parallel computation of all ACS units and the per-symbol dynamic adjustment of *T*. In the implemented decoder, the expected symbol value (*BMselect*) is used to select the appropriate branch metric from the BMG, as shown at the left in Figure 3. This branch metric value is combined with the path metric of its parent present state to form a new path metric, *di*.

The Add-Compare-Select (ACS) unit, shown in detail in Figure

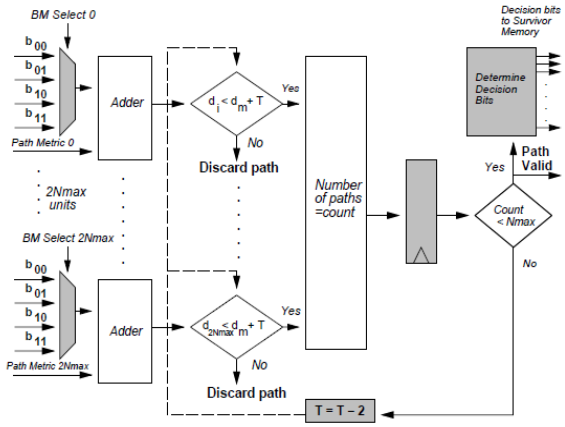

Figure 3 ACS unit of adaptive Viterbi decoder

At each trellis stage, the minimum-value surviving path metric among all path metrics for the preceding trellis stage, *dm*, is computed. New path metrics are compared to the sum *dm* + *T* to identify path metrics with excessive cost. Comparators are then used to determine the life of each path based on the threshold, *T*. If the threshold condition is not satisfied by path metric *dm* + *T*, the corresponding path is discarded.

Once the paths that meet the threshold condition are determined, the lowest-cost *Nmax* paths are selected. Sorting circuitry is eliminated by allowing feedback adjustments to the parameter *T* for each received symbol. If the number of paths that survive the threshold is less than *Nmax*, no iteration is required.

As show in Figure 3, for stages when the number of paths surviving the threshold condition is greater than *Nmax*, *T* is iteratively reduced by 2 for the current trellis stage until the number of paths surviving the threshold condition is equal to or less than *Nmax*. The *T* value is reset to its original value prior to the processing of the next trellis stage. Appropriate values for *T* and *Nmax* were determined in previous work, so that *T* reduction is needed infrequently (for less than 5% of symbols). The output of the ACS units includes *path valid* signals which indicate which of the 2* *Nmax* paths have survived pruning.

C. Survivor Memory Unit (SMU)

SMU design is based on modified register exchange method. A pointer keeps track of the minimum value of the path metric. The survivor memory unit can be designed in two different styles,

1.) Register Exchange (RE)

2.) Trace-Back (TB)

This block is necessary only for the traceback approach. Except for the head and the tail part of the trellis diagram there are two incoming branches for each state. Among the two incoming branches it is necessary to indicate which branch, either upper or lower one, survives. So only one bit of information is necessary for each state to flag the survivor path.

D. Register Exchange (RE)

ISSN 2321 – 2004
ISSN 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 3, June 2013

In general, RE can easily support very high decoding throughput but occupies larger silicon area and tends to consume more power. Assuming each trellis state only has two incoming/outgoing branches, the structure of an RE-based design solution by introducing clock-gating into the RE array followed by majority vote unit. The validity bits from the modified ACS units array directly control the clock-gating for power reduction. The Register Exchange unit for Adaptive Viterbi Decoder shown in Figure 4

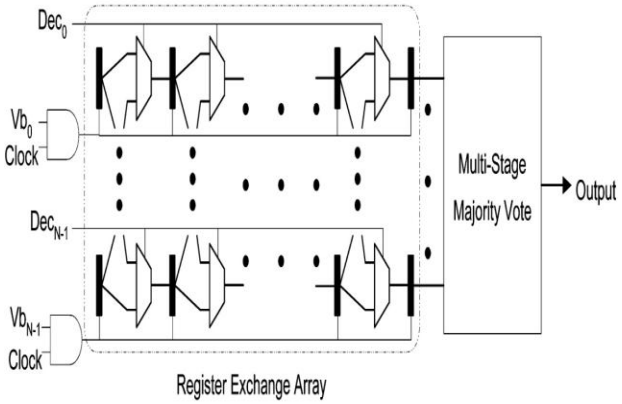

Figure 4 Structure of RE-based design

Since the average number of survivors can be much less than the total number of trellis states in adaptive Viterbi decoding, this may lead to significant power savings. The decoder output is determined by a majority vote unit that only counts decision symbols from survivors.

In this work, we use the multistage majority vote unit design approach. The length of the register exchange array is called decision length. The register-exchange approach assigns a register to each state. The register records the decoded output sequence along the path starting from the initial state to the final state, which is same as the initial state.

The register of state S_1 at $t=3$ contains 101. Note that the trellis follows along the bold path, and the decoded output sequence is 101. This approach eliminates the need to trackback, since the register of the final state contains the decoded output sequence. Hence, the approach may offer a high-speed operation, but it is not power efficient due to the need to copy all the registers in a stage to the next stage.

E. Trace Back (TB)

The trace back module traces back the survivor path after the entire code word has been received and generates the decoded output sequence. It is a combinational block and is active during only one clock cycle as shown below.

To support high throughput in an Adaptive Viterbi Decoder, the TB-based survivor memory unit must provide large enough memory access bandwidth. These techniques can be equally applied to the adaptive Viterbi decoder with only one subtle difference: In the conventional Viterbi decoder, we can start the TB from any arbitrary trellis state; while for the adaptive Viterbi decoder, since not all the trellis states lead to survivors, we need to ensure that TB always starts from a state leading to a survivor in the present decoding depth.

TB requires less silicon area and power but may not readily support very high decoding throughput. In the following, we will discuss how to design the survivor memory unit of a state-parallel Adaptive Viterbi Decoder based on either an RE or TB design style. So Instead of memory units which consume more power due to the decoding of column and row address, registers are used to shift and store the data temporarily.

The TB stores the decisions in an SRAM and reads them in reverse order during the trace-back cycle. As the update rate of the memory is much less frequent than the update rate of the registers in the RE technique, the power consumption is significantly reduced.

V. IMPLEMENTATION RESULTS

There are two approaches for generation of decoded output sequence, register-exchange and traceback. In the register-exchange approach each state has a register to store the survivor path information starting from the initial state to the state. In the traceback scheme, only the survivor path information for each state is stored for each code symbol. Once the complete code word is received, a traceback block extracts the decoded output sequence using the survivor path information.

ISSN 2321 – 2004
ISSN 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 3, June 2013

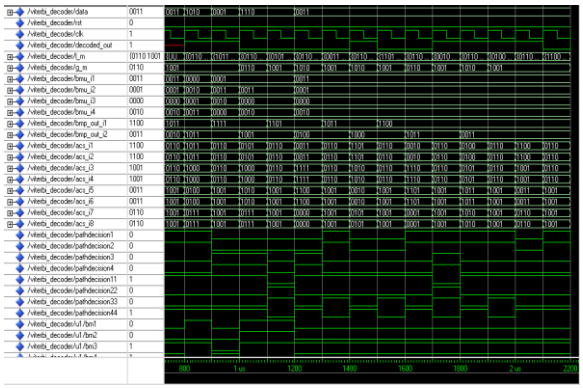

Figure 5 Simulation result for Adaptive Viterbi Decoder

Most communication systems desire links with predictable performance, which is usually specified by a fixed BER. Although desired decoder accuracy remains constant, channel signal-to-noise ratios can vary widely due to factors such as the propagation distance and the shadowing of the transmitted signal by large objects. In the presence of increased noise power (equivalently, a decreased SNR due to a weaker signal), a higher constraint length code is required to maintain a constant BER.

The soft inputs of all VDs are quantized with 7 bits. Each PM in all VDs is quantized as 12 bits. RE scheme with survival length of 42 is used for SMU and the register arrays associated with the purged states are clock-gated to reduce the power consumption in SMU.

VI. CONCLUSION

The use of error-correcting codes has proven to be an effective way to overcome data corruption in digital communication channels. We have proposed a high-speed low-power AVD design for TCM systems. The proposed architecture efficiently reduces the power consumption of AVDs without reducing the decoding speed appreciably. The adaptive Viterbi decoder described has been implemented in Xilinx FPGA architectures for a range of constraint lengths. For a given, fixed bit error and decode rate, power savings is achieved by 19% compared to conventional design with the goal of employing a lower-power decoder. In the future, we plan to consider the decoding benefits of using a hybrid microprocessor and FPGA device. The tight integration of sequential control with parallel decoding may provide further run-time power benefits.

REFERENCES

[1] Jinjin He, Huaping Liu, Zhongfeng Wan, Xinming Huang, and Kai Zhang, "High-Speed Low-Power Viterbi Decoder Design for TCM Decoders" IEEE Transaction Very Large Scale Integration. (VLSI) System volume.20, no.4. April 2012.

[2] "Bandwidth-efficient modulations," Consultative Committee For Space Data System, Matera, Italy, CCSDS 401(3.3.6) Green Book, Issue 1, Apr. 2003.

[3] J. B. Anderson and E. Offer, "Reduced-state sequence detection with convolutional codes," *IEEE Trans. Inf. Theory*, vol. 40, no. 3, pp. 965–972, May 1994.

[4] C. F. Lin and J. B. Anderson, "M-algorithm decoding of channel convolutional codes," presented at the Princeton Conf. Info. Sci. Syst., Princeton, NJ, Mar. 1986.

[5] S. J. Simmons, "Breadth-first trellis decoding with adaptive effort," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 3–12, Jan. 1990.

[6] F. Chan and D. Haccoun, "Adaptive viterbi decoding of convolutional codes over memoryless channels," *IEEE Trans. Commun.*, vol. 45, no. 11, pp. 1389–1400, Nov. 1997.

[7] R. A. Abdallah and N. R. Shanbhag, "Error-resilient low-power viterbi decoder architectures," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4906–4917, Dec. 2009.

[8] J. Jin and C.-Y. Tsui, "Low-power limited-search parallel state viterbi decoder implementation based on scarece state transition," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 11, pp. 1172–1176, Oct. 2007.

[9] F. Sun and T. Zhang, "Low power state-parallel relaxed adaptive viterbi decoder design and implementation," in *Proc. IEEE ISCAS*, May 2006, pp. 4811–4814.

[10] J. He, H. Liu, and Z. Wang, "A fast ACSU architecture for viterbi decoder using T-algorithm," in *Proc. 43rd IEEE Asilomar Conf. Signals, Syst. Comput.*, Nov. 2009, pp. 231–235.

[11] J. He, Z. Wang, and H. Liu, "An efficient 4-D 8PSK TCM decoder architecture," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 5, pp. 808–817, May 2010.