M. Pushpa Latha* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.4, Issue No.6, October – November 2016, 4762-4766.

Design Of High Speed &Low Power Viterbi Decoder For Treelis Coded Modulation(TCM)

M.PUSHPA LATHA
PG Scholar
Dept of ECE
Global College of Engineering & Technology

P.SURESH
Assistant Professor
Dept of ECE
Global College of Engineering & Technology

Abstract: High-speed, low-power design of Viterbi decoders for trellis coded modulation (TCM) systems is presented in this paper. It is well known that the Viterbi decoder (VD) is the dominant module determining the overall power consumption of trellis coded modulation decoders. We propose a pre-computation architecture incorporated with T-algorithm for Viterbi decoder, which can effectively reduce the power consumption without degrading the decoding speed much. A general solution to derive the optimal pre-computation steps is also given in the paper. Implementation result of a VD for a rate-3/4 convolutional code used in a TCM system shows that compared with the full trellis VD, the pre-computation architecture reduces the power consumption by as much as 70% without performance loss, while the degradation in clock speed is negligible.

I. INTRODUCTION

In telecommunication, trellis modulation (also known as trellis coded modulation, or simply TCM) is a modulation scheme which allows highly efficient transmission of information over band-limited channels such as telephone lines. Trellis modulation was invented by Gottfried Ungerboeck working for IBM in the 1970s, and first described in a conference paper in 1976;.

This idea was to group the symbols in a tree like fashion then separate them into two limbs of equal size. At each limb of the tree, the symbols were further apart.

Although hard to visualize in multi-dimensions, a simple one dimension example illustrates the basic procedure. Suppose the symbols are located at [1, 2, 3, 4, ...].. He next described a method of assigning the encoded bit stream onto the symbols in a very systematic procedure. Once this procedure was fully described, his next step was to program the algorithms into a computer and let the computer search for the best codes.

II. CONVOLUTIONAL ENCODER

Convolutional coding has been used in communication systems including deep space communications and wireless communications. Convolutional codes offer an alternative to block codes for transmission over a noisy channel. Convolutional coding can be applied to a continuous input stream. A convolutional encoder is a Mealy machine, where the output is a function of the current state and the current input. It consists of one or more shift registers and multiple XOR gates. XOR gates are connected to some stages of the shift registers as well as to the current input to generate the output.

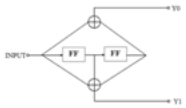

Fig 1: Convolutional Encoder

The encoder in fig1 produces two bits of encoded information for each bit of input information, so it is called a rate 1/2 encoder. A convolutional encoder is generally characterized in (n, k, m) format with a rate of k/n, where

-- N is number of outputs of the encoder

-- K is number of inputs of the encoder

-- M is number of flip-flops of the longest shift register of the encoder

The stream of information bits flows in to the shift register from one end and is shifted out at the other end. The location of stages as well as the number of memory elements determines the minimum hamming distance. Minimum Hamming distance determines the maximal number of correctable bits. Interconnection functions for different rates and different number of memory elements and their minimum hamming distances are available.

The operation of a convolutional encoder can be easily understood with the aid of a state diagram. The state diagram is a graph of the possible states of the encoder and the transitions from one state to another state.

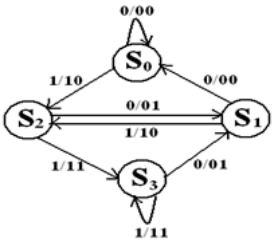

Fig 2: State Diagram for the Convolutional Encoder

Fig 2 represents the state diagram of the encoder shown in Fig 1.Fig 2 depicts state transitions and the corresponding encoded outputs. As there are two memory-elements in the circuit, there are four possible states that the circuit can assume. These four states are represented as S0 through S3. Each state's information (i.e. the contents of flip-flops for the state) along with an input generates an encoded output code. For each state, there can be two outgoing transitions; one corresponding to a '0' input bit and the other corresponding to a '1' input

64-bit convolutional encoder:

We have concluded that two-step precomputation is the optimal choice for the rate-3/4 code VD. For convenience of discussion, we define the left-most register in Fig 3 as the most-significant-bit (MSB) and the right-most register as the least-significant-bit (LSB).

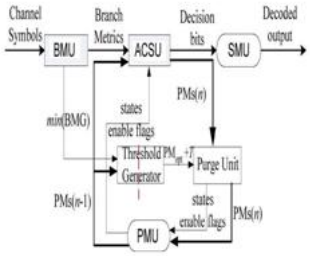

Fig 3: Rate 3/4 Convolutional Encoder

III. IMPLEMENTATION

Working of Viterbi Algorithm:

The major tasks in the Viterbi decoding process are as follows:

1. Branch metric computation
2. State metric update
3. Survivor path recording
4. Output decision generation

Block Diagram of Decoder With Encoder:

There are three major components in viterbi decoder, the branch metric unit (BMU), Add-compare-select unit (ACS), survivor memory unit (SMU) or Trace Back (TB) are shown in the fig 4.

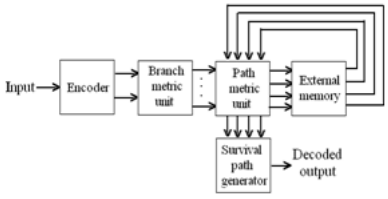

Fig 4: Block diagram of viterbi decoder

Block diagram of two step precomputation architecture:

Branch Metric Unit:

The branch metric unit calculates the branch metrics of the trellis structure from bit metrics as shown in the below fig 6. The branch metrics are difference values between received code symbol and the corresponding branch words from the encoder trellis. The bit metrics can be calculated with a separate unit as shown in figure or a look-up table can be used. The inputs needed for this task are bit metrics, which in this case come from the convolutional encoder. These encoder branch words are the code symbols that would be expected to come from the encoder output as a result of the state transitions. In hard-decision decoding the calculation method is called Hamming distance. The Hamming distance d(X, Y) between two words X and Y is defined to equal the number of differing elements. For soft-decision decoding, there is another algorithm called Euclidean distance. When the input symbol is X and encoder symbol is Y, the Euclidean distance is calculated from the formula $(X-Y)^2$.

Fig 6: Branch Metric computation Block

Path Metric Unit:

The PMU calculates new path metric values and decision values as shown in the fig 7 below. Because each state can be achieved from two states from the earlier stage, there are two possible path metrics coming to the current state. The ACS unit, as shown in figure, adds for each of the two incoming branches the corresponding states path metric, resulting in two new path metrics. A new value of the state metrics has to be computed at each time instant. In other words, the state metrics have to be updated every clock cycle. Because of this recursion, pipelining, a common approach to increase the throughput of the system, is not applicable. The Add-Compare-Select (ACS) unit hence is the module that consumes the most power and area. In order to obtain the required precision, a resolution of 7 bits for the state metrics is essential, while 5 bits are needed for the branch metrics. Since the state metrics are always positive numbers and since only positive branch metrics are added to them, the accumulated metrics would grow indefinitely without normalization. In this project we have chosen to implement modulo normalization, which requires keeping an additional bit. The operation of the ACS unit is shown in figure. The new branch metrics are added to previous state metrics to form the candidates for the new state metrics. The comparison can be done

by using the subtraction of the two candidate state metrics, and the MSB of the difference points to a larger one of two.

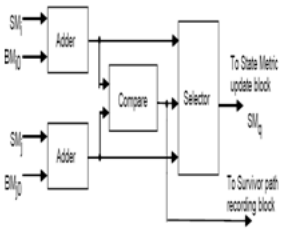

Fig 7: Path metric unit

The two possible states in Trellis diagram, and if the decision bit is zero the path metric selected is coming from the upper state. As the ACS unit needs the results from the calculations of the previous steps, it forms a feedback loop with the external memory unit, where the results are stored.

Survival Path Unit:

The survivor path unit stores the decisions of the ACS unit and uses them to compute the decoded output. The trace-back technique and the register-exchange approaches are two major techniques used for the path history management .The former takes up less area but require much more time than the latter, since it needs to search the trace of the survivor path back sequentially. A relatively new approach called permutation network based path history unit implements directly the trellis diagram of the given convolutional code. The resulting circuit has smaller routing area than register-exchange technique and has faster decoding speed than trace-back method regardless of the constraint length.

In order to decode the input sequence, the survivor path, or shortest path through the trellis must be traced. The selected minimum metric path from the ACS output points the path from each state to its predecessor. In theory, decoding of the shortest path would require the processing of the entire input sequence. In practice the survivor paths merge after some number of iterations, as shown in bold lines in the 4-state example of figure. From the point they merge together, the decoding is unique. The trellis depth at which all the survivor paths merge with high probability is referred as the survivor path length.

IV. VITERBI ALGORITHM

Modules Used In Implementation

1) Trellis codec:

The top module consisting of convolutional encoder and all the modules of the viterbi decoder and the decoded sequence of data with input as the message bits.

2) Convolutional encoder:

We have used a rate ½ convolutional encoder. Hence, it uses a state machine of four states and generates 2 bit output with 1 bit input.

3) Branch metric unit:

This module computes the metric for each path in the trellis. We have used a look-up table which gives the output branch metric values, based on the encoded sequence

4) Path metric unit:

This module computes the new path metric value by summing up the incoming path metric and branch metric values. It also compares the two metrics generated at each stage and selects the minimum of the two as the new path metric value for that particular state. Based on the path selected, it generates the selection bits which are used to traceback the original path by the survivor path unit. A path metric memory is used to store the path metric values. This forms a feedback loop with path metric unit.

Algorithmic Flow of Implementation:

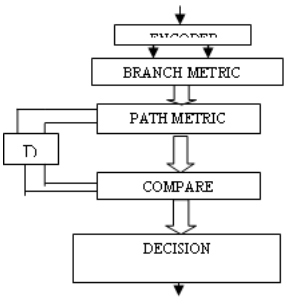

Fig 8: Viterbi Algorithm

V. RESULTS

Simulation Results:

The simulations results are taken from the XILINX tool and the below shown fig 9 describes the simulation results of convolutional encoder and the output will be according to the states in the convolutional encoder.

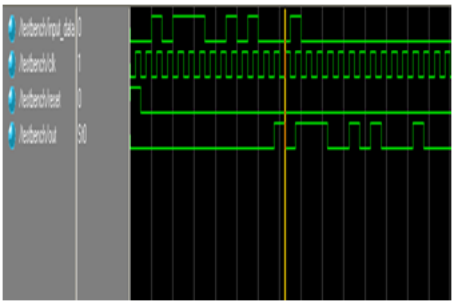

Fig 9: Simulationl result of convolutional encoder

Simulation Results Of Viterbi Decoder

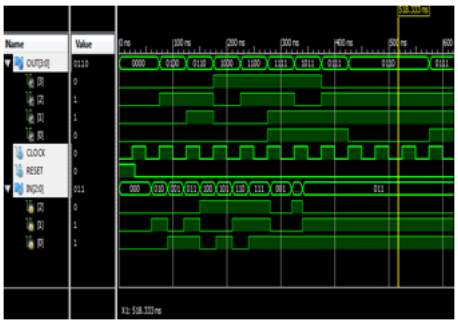

Fig 10: Simulation result of Viterbi Decoder

These wave forms are corresponding to the data which we have given at the encoder input. we know that in any digital communication channel the output data at the decoder is same as that of the input data that was given at the encoder input. So that, the output which we got at the viterbi decoder is same as that of the input which we have given at the input of the encoder. observe in the output wave forms shown in above fig 10.

VI. SYNTHESIS REPORTS

The following table 1 consist of different reports i.e., speed and timing and their comparisons both proposed system and existing system.

Comparison Table

Table 1: Synthesis Reports of Viterbi Decoder

COMPARISION TABLE		
	Max SPEED(MHZ) of Propsed system	Max speed(MHZ) of Existing system
VD with 2-step Precomputation	468.165	103.8
Convolution enoder	1448.646	88.2
TIMING	2.136ns	14.26ns

VII. CONCLUSION

We have proposed a high-speed low-power VD design for TCM systems. The pre-computation architecture that incorporates T –algorithm efficiently reduces the power consumption of VDs without reducing the decoding speed appreciably. We have also analyzed the pre-computation algorithm, where the optima pre-computation steps are calculated and discussed. Both the ACSU and SMU are modified to correctly decode the signal.

Future Scope

The Viterbi Decoder is designed for 64-bits so the design is going to be complex. So, to overcome this problem in future, the Reed-Solomon codes are introduced and these are going to be in the form of bytes and this will reduces the design complexity.

VIII. REFERENCES

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

[2] www.ijsrt.com

[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, "T-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, "Lowpower 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.

[12] www.synopsys.com

[13] www.iasir.net

[14] www.complextoreal.com

AUTHOR's PROFILE

M.Pushpa Latha, Pursuing M.Tech(VLSI), Global Institute Of Engineering &Technology, B.Tech (Electronics & Communication Engineering) At Gouthami Institute Of Technology&Management For Women

P.Suresh M.Tech (Embedded Systems),Assistant Professor & Head Of The Department Of ECE At Global College Of Engineering & Technology, Kadapa, AP, India.

