VLSI Implementation of Error Correction Unit for TCM Decoders Using T-Algorithm

S. Sathyabama and S.D. Jayavathi

***Abstract*—*A design of Viterbi decoder for Trellis Coded Modulation (TCM) systems is implemented to reduce the power and to increase the speed efficiently. In the existing work, Reed Solomon (RS) codes are widely employed for error correction. The complexity of RS encoder and decoder increases with the error correcting capability of the codes. So it requires more hardware and power. But VLSI designer always prefer to have simple and cascadable structure with the interconnection for reliable and high speed operation of the circuit. For this, Viterbi decoder (VD) based byte error correcting code has been designed. Viterbi decoder based scheme can be easily extended for correcting more than two byte errors. These require less hardware compared to existing techniques of RS codes and it determines the overall power consumption of TCM decoders. A precomputation architecture incorporated with T-algorithm for Viterbi decoder, which can effectively reduce the power consumption without degrading the decoding speed much. The whole proposed system will be designed using VHDL and simulation is performed by xilinx software. This synthesized system will be tested against FPGA Spartan 3E for the improved performance evaluation.*

***Indexterms*—** *Reed Solomon, Trellis Coded Modulation, Viterbi decoder, VLSI*

I. INTRODUCTION

TRELLIS coded modulation (TCM) is a technique that combines error-correcting coding and modulation in digital communications. The scheme gains noise immunity over un-coded transmission without changing the data rate. In a power limited environment, the desired system performance should be achieved with the smallest possible transmitted power. The error correcting codes can increase power efficiency by adding extra bits to the transmitted symbol sequence. This requires the modulator to operate at a higher data rate, which requires a wider bandwidth. From this, a large signal power would be required to maintain the same system Bit Error Rate (BER). In order to achieve improved reliability of a digital transmission system without increasing transmitted power or required bandwidth, both coding and modulation are considered in TCM technology. TCM also improves system reliability without increasing transmitting power and required channel bandwidth.

In this, a TCM system employs a high-rate convolutional code, which leads to a high complexity of the Viterbi decoder (VD) for the TCM decoder, even if the constraint length of the convolutional code is moderate. For example, rate-3/4 convolutional code used in a 4-D TCM system for deep space communications has a constraint length of 7, thus the computational complexity of the corresponding viterbi decoder is equivalent to that of rate-1/2 convolutional code with a constraint length of 9 due to the large number of transitions in the trellis. Thus, Viterbi decoder is the dominant module determining the overall power consumption of TCM decoders.

In the existing work, low power VD design has been well studied. Power reduction in Viterbi decoder can be achieved by reducing the number of states (RSSD), M-algorithm. RSSD reduces the state trellis of a channel code by forming the states into classes. RSSD cannot be more efficient than the M-algorithm. However, M-algorithm requires sorting process at the feedback loop. Using this algorithm, it cannot easily recover the lost transmitted path, which slightly reduces the effective data throughput. Furthermore, when the correct path is lost, the bit error probability of the M-algorithm is quite large due to error propagation. Finally, T-algorithm is more commonly used for searching the optimal path metric (PM) and very efficient in reducing the power consumption.

In this paper, an add compare select unit (ACSU) architecture based on precomputation for Viterbi decoders incorporating T-algorithm, which efficiently improves the clock speed of Viterbi decoder. We further analyze the precomputation algorithm. A systematic way to determine the optimal precomputation steps is presented, where the minimum number of steps for the critical path to achieve the theoretical iteration bound is calculated and the computational complexity overhead due to precomputation is evaluated.

II. VITERBI DECODER

The functionality of a Viterbi decoder is shown in fig. 1. First, the branch metric unit (BMU) calculates distance (metric) between the received noisy symbol and the output symbol of the state transition. Then the branch metrics are fed into the add compare select unit (ACSU) that recursively computes the path metric. ACSU computes the accumulated metric associated with the sequence of transitions(path) to reach a state. ACSU selects the path with lowest metric value, which is the survivor path. Survivor Memory Unit (SMU)

S. Sathyabama, PG Scholar, Department of ECE, National Engineering College, Kovilpatti, Tuticorin, Tamil Nadu. E-mail: ssb.bama@gmail.com

S.D. Jayavathi, Assistant Professor (SG), Department of ECE, National Engineering College, Kovilpatti, Tuticorin, Tamil Nadu.

stores the information that permit to trace back from a state to the previous one.

A. Figures

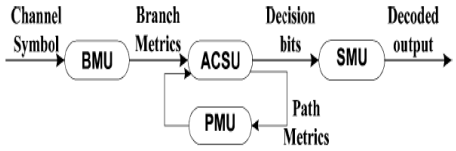

Figure 1: Functional Diagram of a Viterbi decoder

. Important definitions

The following terms are vital to the understanding of convolution codes.

1) Hard decision decoding: A decoder which receives only bits from the channel (without any reliability scheme) and a hamming distance used as a metric.

2) Soft decision decoding: A decoder which receives a bit stream containing information about reliability of each received symbol.

3) Code rate R=(k/n): Number of bits into convolutional encoder(k)/ number of bits in output symbol which corresponds not only current input bit, but also previous (K-1) ones.

4) Constraint Length K: It denotes the "length" of the convolutional encoder, i.e., number of k-bit stages are available to feed the combinational logic that produces the output symbols.

III. Low Power High Speed Viterbi Decoder Design

Viterbi decoder is designed in a Trellis Coded Modulation scheme using Conventional T-algorithm. This algorithm can reduce the computational complexity by reducing the number of iterations. By using pre-computational logic along with conventional T-algorithm, better performance in Bit Error Rate (BER) is achieved. By improving the decoding speed, the power usage was automatically reduced. Error correction rate is improved. The main purpose of the work is, Forward Error Correction (FEC) which can improve the capacity of the channel. Convolution codes offer best FEC which blocks the codes over a noise transmission, here Viterbi maximum likelihood algorithm one of the best technique for wireless communication FEC adds some redundancy with data while transmission over a noisy channel. To achieve high speed and low power using maximum likelihood algorithm condition, there is a need to design an efficient Viterbi decoder. Latency is reduced by computing ACS condition and the M-step look ahead is to be designed and it improves the hardware efficiency. To avoid computational complexity a new scheme, 'Pre computational (T-algorithm)' is used. Numbers of iteration bounds are reduced by using this pre-computation logic to avoid computation complexity. By using this algorithm, low power simplified architecture design is obtained. Thereby, performance is improved, the response time is effective, and the error correction rate is better.

A. Rate ½ Convolutional Encoder

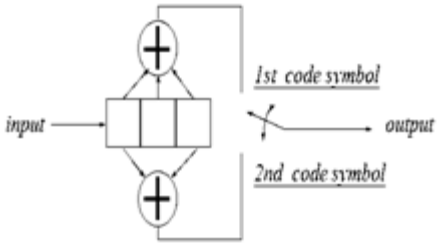

Figure 2 : ½ Convolutional Encoder

Constraint length, K =T+1=3 (Shift registers, T=2)
Number of states, $2^{K-1}=2^{3-1}=2^2=4$ states
Rate= Convolutional code rate (X/Z)
X=Number of input bit
Z=Number of output bits

These are three parameters which define the convolutional code:

1) Rate: Ratio of the number of input bits to the number of output bits. In this example, rate is 1/2 which means there are two output bits for each input bit.

2) Constraint length: The number of delay elements in the convolutional coding. In this example, with K=3 there are two delay elements.

3) Generator polynomial: Wiring of the input sequence with the delay elements to form the output. In this example, generator polynomial is $[7,5]_8 = [111,101]_2$. The output from the $7_8=111_2$arm uses the XOR of the current input, previous input and the previous to previous input. The output from the $5_8=101_2$uses the XOR of the current input and the previous to previous input.

B. Internal Architecture Of Viterbi Decoder

There are three major components in Viterbi decoder, the Branch Metric Unit (BMU), Add-Compare-Select Unit (ACSU), Survivor Memory Unit (SMU) or Trace Back (TB).

1. Branch Metric Unit (BMU)

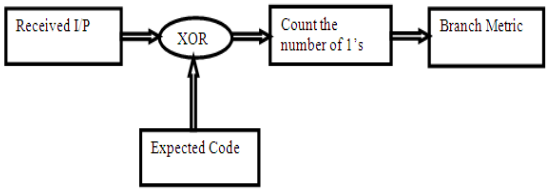

Figure 3: Branch Metric Unit

The Branch Metric Unit calculates the branch metrics of the trellis structure from bit metrics. The branch metrics are difference values between received code symbol and the corresponding branch words from the encoder trellis. 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. Consider two received coded bits at a time

y_i, and compute the hamming distance between all possible combinations of two bits. The number of differing bits can be computed by XORing y_i with 00,01,10,11 and then counting the number of ones.

$hd_{i,00}$ is the number of 1′s in $00 \oplus y_i$
$hd_{i,01}$ is the number of 1′s in $01 \oplus y_i$
$hd_{i,10}$ is the number of 1′s in $10 \oplus y_i$
$hd_{i,11}$ is the number of 1′s in $11 \oplus y_i$

2. Add Compare Select Unit (ACSU)

The Add Compare Select Unit selects the path with lowest metric value and that path which stored in the survivor path.

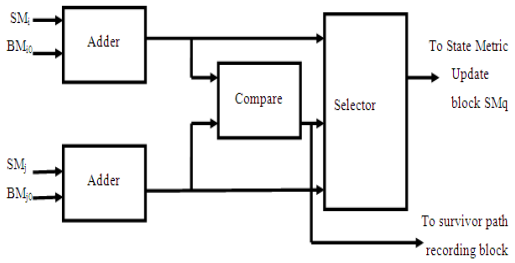

Figure 4: Add Compare Select Unit

The Add-Compare-Select Unit (ACSU) is the heart of the Viterbi algorithm and calculates the state metrics and also recursively accumulates the branch metrics as the path metrics, compares the incoming path metrics, and makes a decision to select the most likely state transitions for each state of the trellis and generates the corresponding decision bits. The path metrics are added to state metrics from the previous time instant and the smaller sum is selected as the new state metric:

$SM1_n = \min (SM1_{n-1} + bm_1, SM2n_{-1} + bm_3)$
$SM2_n = \min (SM1_{n-1} + bm_2, SM2n_{-1} + bm_4)$

bm_k is the hamming distance between received and expected sequence.

3. Survivor Memory Unit (SMU)

The Survivor Memory Unit (SMU) is a memory, storing the decision bits for each state, each time step and the previous state. SMU is responsible for keeping track of the information bits associated with the surviving paths designated by the path metric updating and storage unit.

There are two basic design approaches for SMU:

- Register Exchange
- Trace Back

In both techniques, a shift register is associated with every trellis node throughout the decoding operation. This register has a length equal to the frame length. The Register exchange method works well for small constraint lengths. The Trace back method works well for longer constraint length codes. The Trace back method stores the decisions from the ACS into a RAM and also the path information in the form of an array of recursive pointers. The best path is determined by reading backwards through the RAM. It is important to note that the trace back block can be activated only at the end of each codeword and deactivated for the rest of period. This fact is utilized to reduce the switching activities inside the module and the power dissipation. In Trace back approach, each registers storing the survivor path information updates its contents only once, during the entire period of a codeword. In contrast, Register exchange approach, all registers update their contents for each code symbol. Hence, the registers in trace back module would dissipate less power. In practice, the survivor paths merge after some number of iterations. The trellis depth at which all the survivor paths merge with high probability is referred to as the survivor path length.

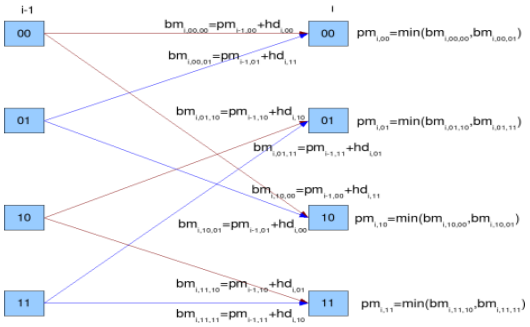

Figure 5: State Diagram for Branch metric and Path Metric Computation

IV. PROPOSED PRE-COMPUTATION ARCHITECTURES

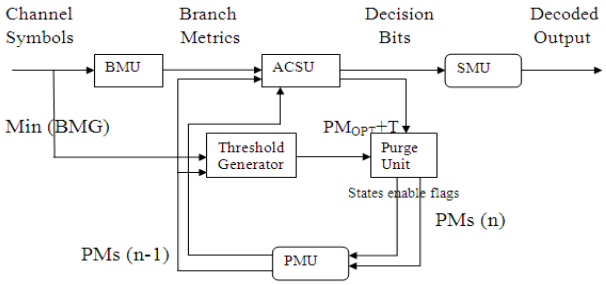

Figure 6: Viterbi Decoder with Two Step Precomputation T-Algorithm

The Functional block diagram of the VD with two-step Pre-computation T-algorithm is shown in Figure.6. The minimum value of each BM Group (BMG) can be calculated in BMU or TMU and then passed to the 'Threshold Generator' unit (TGU) to calculate (PM_{opt}+T). (PM_{opt}+T) and the new PMs are then compared in the 'Purge Unit' (PU).

V. IMPLEMENTATION RESULTS

The Convolutional Encoder is simulated using Xilinx software. The simulation result is shown in figure 7 respectively.

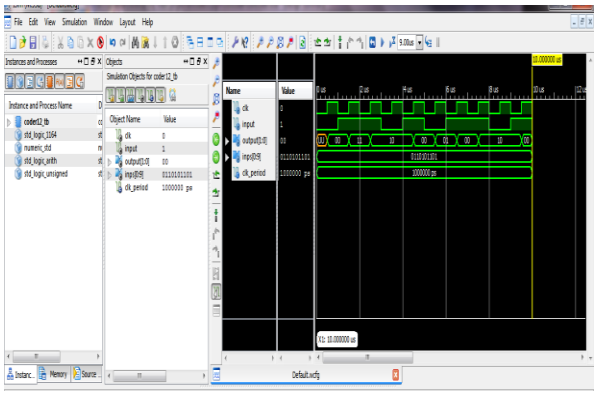

Figure 7: Output for Convolutional Encoder

The Viterbi decoder is simulated using Xilinx software. The simulation result is shown in figure 8 respectively.

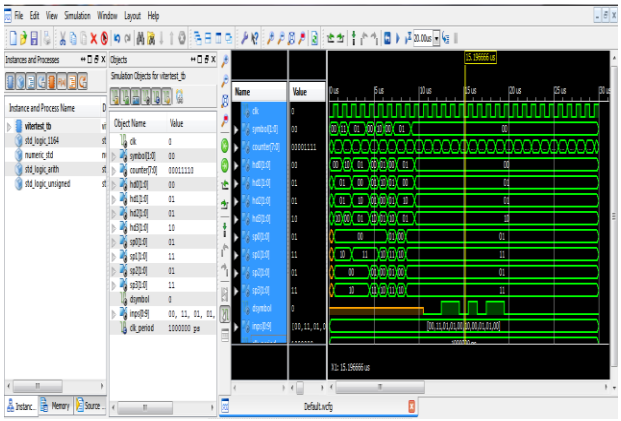

Figure 8: Output for Viterbi Decoder

VI. CONCLUSION AND FUTURE WORK

From the results shown above, it is inferred that convolutional encoder and viterbi decoder was designed for four states and its output was verified using Xillinx ISE tool. We have also analyzed the precomputation algorithm, where the optimal precomputation steps are calculated and analyzed. In future, the precomputation architecture that incorporates T-algorithm efficiently reduces the power consumption of viterbi decoder without reducing the decoding speed has to be verified by analyzing the parameters.

REFERENCES

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

[2] Chao Cheng and Keshab parthi.K (2008), 'Hardware Efficient Low-Latency Architecture for High Throughput Rate Viterbi Decoders,' IEEE Trans on circuits and systems, vol.55, no. 12.

[3] Chan.F and Haccoun.D (1997), 'Adaptive viterbi decoding of convolutional codes over memory less channels,' IEEE Trans. Commun., vol. 45, no. 11, pp. 1389–1400.

[4] Fei Sun and Tong Zhang (2005), 'Parallel High-Throughput Limited Search Trellis decoder VLSI design,' IEEE Trans .on very large scale integration (VLSI) systems, vol, 13, no.9.

[5] He.J, Liu (2011), 'Reconfigurable Efficient Design Of Viterbi Decoder for Wireless Communication System,'IEEE Trans. Very Large Scale Integr.(VLSI) Syst.,Vol.2,No. 7.

[6] Jinjin He and Huaping Liu (2012), 'High speed Low power Viterbi decoder Design for TCM decoders,' IEEE Trans. Very Large Scale Integr. (VLSI)Syst., vol. 20, no.4.

[7] Jin.J and Tsui.C.Y (2007), '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.

[8] Jinjin He and Zhongfeng Wang (2010), 'An Efficient 4-D 8PSK TCM Decoder Architecture,' IEEE Trans. Very Large Scale Integr (VLSI) Syst., vol. 18, no. 5.

[9] Lin.C.F and Anderson.J.B (1986), 'M-Algorithm decoding of channel Convolutional codes,' presented at the Princeton Conf. Info. Sci. Syst., Princeton, NJ.

[10] Rami Abdallah.A and Shanbhag.R (2009), ' Error-Resilient Low-Power Viterbi Decoder Architectures,' IEEE Trans.on signal processing., vol.57, no.12.

[11] Simmons.S.J (1990), 'Breadth-first trellis decoding with adaptive effort,' IEEE Trans. Commun., vol. 38, no. 1, pp. 3–12.

[12] Seshagiri Rao.P (2012), ' Viterbi Decoder With low power and low complexity for Space Time Trellis Code,' vol. 2, Issue 3,pp.1359-1365.

[13] Zhang.T and Sun.F (2006), 'Low power state-parallel relaxed adaptive Viterbi decoder design and implementation,' in Proc. IEEE ISCAS, pp. 4811–4814.

ISBN 978-93-82338-53-6 | Published by Bonfring