

Implementation of Viterbi Decoder using T-algorithm for TCM Decoders

Valiveti.Ganesh Kumar¹, A.Ch.Sudhir²

M.Tech VLSI Design, Department of ECE, Gitam University, Visakhapatnam, India¹

Assiatant Professor, Department of ECE, Gitam University, Visakhapatnam, India²

Abstract: Viterbi Decoder employed in digital wireless communication plays a dominant role in overall power consumption of trellis coded modulation (TCM) decoder. Using T-algorithm dramatically reduces the decoding speed. The key point in improving clock speed of T-algorithm is to quickly find the optimal Path Metric ($PM_{optimal}$). The proposed Viterbi Decoder (VD) implementation can reduce the power consumption with less reduction in the maximum decoding speed. The trellis path in the circuit is found using trellis coded modulation. Pre-computation technique has been adopted for the trellis coded modulation. This paper focuses on the realization of convolutional encoder and Viterbi decoder with a constraint length (K) of 7 and code rate (k/n) of 3/4 using Field Programmable gate array (FPGA) Technology. The convolutional encoder and Viterbi decoder (VD) using T-algorithm has been coded in Verilog HDL code and implemented in Xilinx ISE 14.2

Keywords: Trellis coded modulation (TCM), Viterbi Decoder (VD), Field Programmable gate array (FPGA), Path Metric ($PM_{optimal}$).

I. INTRODUCTION

In Today's digital communications reliability and efficiency are the most concerned factors while transmitting data through communication channels. Error correction technique plays an important role in communication systems.

Convolutional encoding with Viterbi Decoding (VD) using Viterbi algorithm can be used as a Forward error correction technique and this approach provides good performance with low cost and is particularly suited to a channel in which the transmitted signal is corrupted mainly by Additive White Gaussian Noise (AWGN).

Trellis coded modulation (TCM) employs a high rate convolutional code as they are used in bandwidth-efficient systems. Though the constraint length of the convolutional encoder is moderate complexity is high for Viterbi decoder (VD) with Trellis Coded Modulation (TCM) decoder. Viterbi Decoder has quite high Power consumption due to the large number of transitions in the trellis diagram.

In a TCM decoder Low Power schemes should be developed for the VD, In order to reduce the power consumption as well as the computational complexity. T-algorithm showed efficiency in reducing the power consumption. Search of optimal PM in the feedback loop can still reduce the decoding speed. Two types of variations have been proposed for T-algorithm the first is the relaxed adaptive VD where instead of finding the real one in each cycle estimated optimal Path Metric is suggested and the limited-search parallel state VD based on scarce state transition (SST) [3]. We proposed an add-compare-select unit (ACSU) architecture based on

precomputation incorporating T-algorithm for Viterbi Decoder can efficiently improve clock speed.

II. CONVOLUTIONAL ENCODING WITH VITERBI DECODING

A. Convolutional Encoder

A convolutional encoder is a type of error-correcting code technique which contains memory and the n encoder outputs at any given unit of time depend not only on the k inputs at that time unit but also on m previous input blocks.

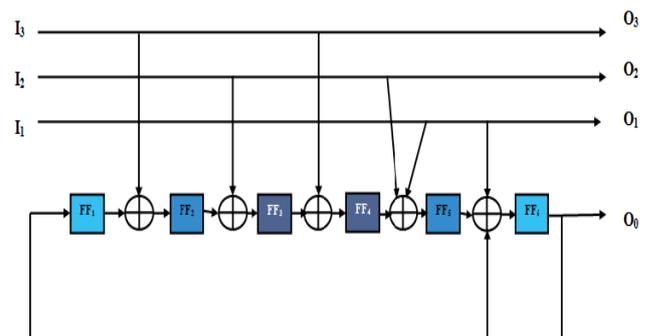


Figure 1: Convolutional Encoder with K=7 and k/n =3/4

Convolutional codes are usually characterized by two parameters code rate (k/n) and constraint length (K) and the patterns of n modulo-2 adders. The shift register has a constraint length (K) of 7; equal to the number of stages in the register the encoder has n generator polynomials, one for each adder. Input bits are fed to I1, I2, and I3. Using the generator polynomials and the existing values in the remaining registers, the encoder outputs O0, O1, O2, O3. O1, O2, O3 takes the input bits I1, I2, I3 without any change. The key role is played by the O0 which is exhibited with a delay after performing xor operation at the internal stages.

B. Choosing Precomputation Steps :

In a TCM system, the convolutional code usually has a coding rate of $R/R+1$ and the logic delay of the ACSU is $T_{ACSU} = T_{adder} + T_{p-in-comp}$. If T-algorithm is implemented in the VD, because of a two-input comparator in loop the iteration bound is slightly longer than TACSU to compare the new PMs with a threshold value obtained from the optimal PM and a preset T and is given by

$$T_{bound} = T_{adder} + T_{p_in_comp} + T_{2_in_comp} \quad (5)$$

where T_{adder} is the logic delay of the adder to compute PMs of each candidate path that reaches the same state and $T_{p-in-comp}$ is the logic delay of a p-input comparator (where $p=2^R$) to determine the survivor path for each state. q-step precomputation can be pipelined into q stages, where as q increases the logic delay of each stage is continuously reduced which improves the decoding speed of the low-power VD. After reaching a certain number of steps, q_b precomputation further would not result in additional benefits because of the inherent iteration bound of the ACSU loop.

To achieve the iteration bound expressed in (5), we limit the comparison to be among only p or 2p metrics for the precomputation in each pipelining stage. If each stage reduces the number of the metrics to $1/p$ (or $2-R$) of its input metrics. The iteration bound should satisfy the below conditions with the smallest number of precomputation steps (q_b). $[(2^R)^{q_b}] \geq 2^{(k-1)}$. Therefore $q_b \geq (k-1)/R$ and we express this as $q_b = \lceil (k-1)/R \rceil$ where $\lceil \cdot \rceil$ denotes ceiling function.

Computational overhead is a key factor that should be carefully evaluated. If there are m remaining metrics after comparison at any point of stage, the computational overhead from that stage is at least m addition operations.

For a code with a constraint length k and precomputation steps q, the number of metrics will reduce and the overall computational overhead is

$$N_{overhead} = 2^0 + 2^{(k-1)/q} + 2^{2(k-1)/q} + \dots + 2^{(q-1)(k-1)/q} \quad (6)$$

The estimated computational overhead increases exponentially to q. In real time design, the overhead increases even faster. Therefore, a small number of pre computational steps is preferred even though the iteration bound may not be fully satisfied. One- or two-step precomputation is a good choice in most cases. For TCM systems, where high-rate convolutional codes are always implemented with two steps of precomputation which could achieve the iteration bound and also the computational overhead is reduced.

IV. VITERBI DECODER DESIGN FOR HIGH SPEED LOW POWER

BER performance of a 4-D 8PSK TCM system [2] [5] with a rate $(k/n) 3/4$ is almost same as the conventional T-algorithm since the precomputation algorithm always finds the accurate optimal PM.

A. ACSU DESIGN :

Convolutional encoder with Rate $(k/n) 3/4$ and length (K) 7 is shown in Figure. 1. For convenience left-most register is defined as the most-significant-bit (MSB) and the right-most register as the least-significant-bit (LSB). The 64 states and PMs are labelled from 0 to 63. ACSU (7) [1] feedback loop with two-step pre-computation is expressed as

$$PM_{optimal}(n) = \text{Min} \left\{ \begin{aligned} &\text{Min (cluster 0 (n-2) + min (BMG0 (n-1))),} \\ &\text{Min (cluster 1 (n-2) + min (BMG1 (n-1))),} \\ &\text{Min (cluster 2 (n-2) + min (BMG3 (n-1))),} \\ &\text{Min (cluster 3 (n-2) + min (BMG2 (n-1))) } \\ &\quad + \text{min (even BMs (n))}, \\ &\text{Min \{ min (cluster 0 (n-2) + min (BMG1 (n-1)),} \\ &\text{Min (cluster 1 (n-2) + min (BMG0 (n-1)),} \\ &\text{Min (cluster 2 (n-2) + min (BMG2 (n-1)),} \\ &\text{Min (cluster 3 (n-2) + min (BMG3 (n-1)))} \\ &\quad + \text{min (odd BMs (n))} \end{aligned} \right\} \quad (7)$$

Where

$$\text{Cluster 0} = \{PM_m | 0 \leq m \leq 63, m \bmod 4 = 0\};$$

$$\text{Cluster 1} = \{PM_m | 0 \leq m \leq 63, m \bmod 4 = 2\};$$

$$\text{Cluster 2} = \{PM_m | 0 \leq m \leq 63, m \bmod 4 = 1\};$$

$$\text{Cluster 3} = \{PM_m | 0 \leq m \leq 63, m \bmod 4 = 3\};$$

$$\text{BMG0} = \{BM_m | 0 \leq m \leq 15, m \bmod 4 = 0\};$$

$$\text{BMG1} = \{BM_m | 0 \leq m \leq 15, m \bmod 4 = 2\};$$

$$\text{BMG2} = \{BM_m | 0 \leq m \leq 15, m \bmod 4 = 1\};$$

$$\text{BMG3} = \{BM_m | 0 \leq m \leq 15, m \bmod 4 = 3\};$$

The VD with two-step precomputation T-algorithm functional block diagram is shown in Fig. 3. The minimum value of each Branch Metric Group (BMG) can be calculated in BMU or TMU

and then passed to the “Threshold Generator” unit (TGU) to calculate $(PM_{optimal} + T)$. $(PM_{optimal} + T)$ and the new PMs are then compared in the “Purge Unit”(PU). The architecture of the TGU is shown in Figure. 4.

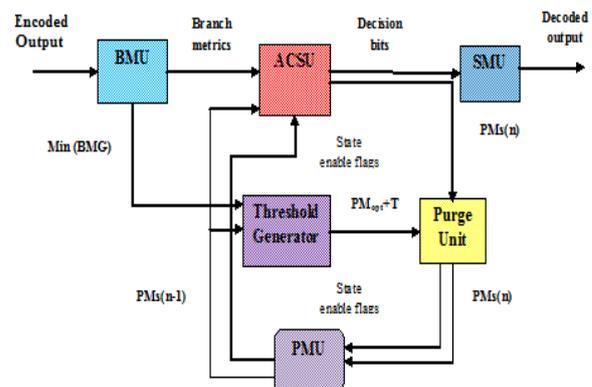


Figure 3: 2-step pre-computation T-algorithm for VD

A. SMU Design:

There are two different types of SMU in the literature: trace back (TB) schemes and register exchange (RE) schemes. Without any low-power schemes in the regular VD, if Register Exchange is used SMU always outputs the decoded data from a fixed state, or traces back the survivor path from the fixed state if TB scheme is used. For VD incorporated with T- algorithm, at all clock cycles no state is guaranteed to be active throughout.

Thus it is impossible to appoint a fixed state for either outputting the decoded bit (Register Exchange scheme) or starting the trace-back process (Trace Back scheme). In the conventional implementation of T-algorithm, the decoder can use the optimal state (state with $PM_{optimal}$) which is always enabled, to output or trace back the data.

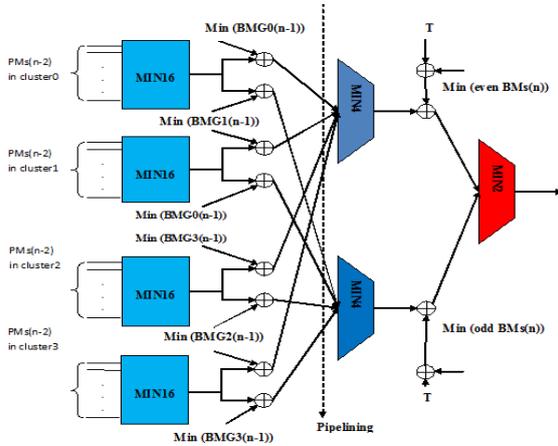


Figure 4: Threshold Generator unit architecture

In the process of searching for the $PM_{optimal}$ it is difficult to find the index of the optimal state, as the estimated $PM_{optimal}$ calculated from PMs at the previous time slot.

It is for this purpose we use 64-to-6 priority encoder [4]. The output would be the unpurged state with the lowest index for 64-to-6 Priority encoder. Assuming the purged states have flag “0” and other states are assigned with the flag “1”. Implementation of such direct 64-to-6 is not easy, so we use of four 4-to-2 priority encoders for the 64-to-6 priority encoder. The same is shown in Figure. 5.

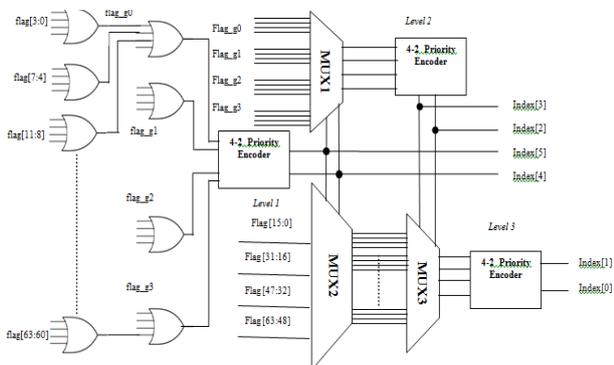


Figure 5: 64-to-6 priority encoder architecture.

V.SYNTHESIS AND SIMULATION RESULT

Viterbi decoder with rate $3/4$ and $K=7$ is realized by FPGA device xc6slx1001-1Lfgg676. The Synthesis Results for Maximum Clock Speed, Power Estimation Results, device utilization summary, logic utilization and distribution report is shown in Table I. The precomputation VD reduced the power consumption by nearly 70% with minimum decoding speed. The VD design is simulated by Xilinx ISE 14.2.

A. Simulation Results:

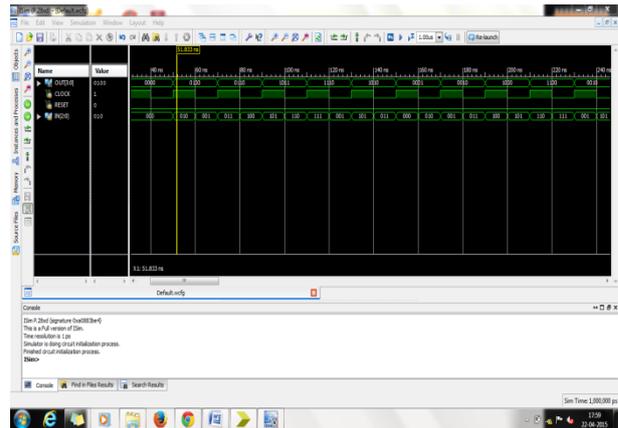


Figure 6: Convolutional Encoder with $(k/n)=3/4$ and $K=7$.

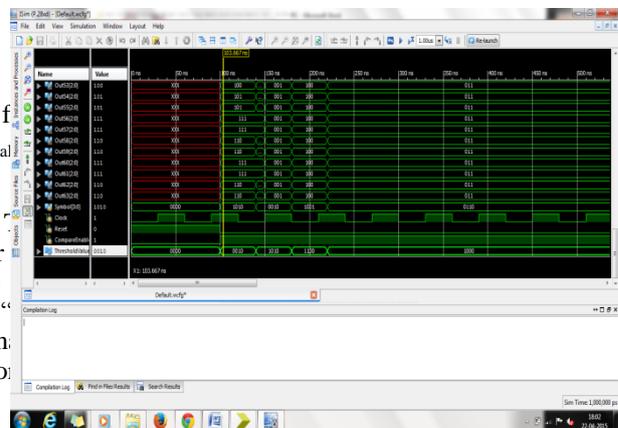


Figure 7: Viterbi Decoder employed by T-algorithm Output.

B. Synthesis Report:

- 1. Power=60mW
- 2. Clock Speed

Minimum period: 8.625ns (Maximum Frequency: 115.944MHz)
Minimum input arrival time before clock: 13.491ns
Maximum output required time after clock: 18.887ns
Maximum combinational path delay: 20.717ns

TABLE I: DEVICE UTILIZATION SUMMARY

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	7373	126576	5%
Number of Slice LUTs	15815	63288	24%
Number of fully used LUT-FF pairs	3893	26150	14%
Number of bonded IOBs	203	480	42%

RADAR and Microwave Engineering in the year 2009. Currently he is perusing PhD in JNTU Kakinada. His area of interests is Wireless communications and Vlsi Design .He Published his research Papers in referred International Journals and International and National Conferences.

VI. CONCLUSION

A high speed low power Viterbi Decoder design for TCM Decoders has been proposed and the Precomputation architecture that uses T-algorithm effectively reduces the power consumption of Viterbi Decoders without reduction in the decoding speed .The same is also analyzed with the optimal precomputation steps calculation. Power Estimation and Synthesis Results

For Maximum clock speed are shown which is reduced by 70% with less reduction in maximum decoding speed. The different modules are designed using Verilog HDL and synthesized using Xilinx Integrated software environment (ISE) 14.2.

REFERENCES

- [1] 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
- [2] 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,pp808-817,May 2010.
- [3] J.Lin and C-Y.Tsui, “Low-power limited- search parallel state Viterbi decoder implementation based on scarce state transition,”IEEE Trans.Very Large Scale Integr.(VLSI)Syst.,vol.15,no.11,pp 1172-1176,Oct.2007.
- [4] J.He, H.Liu,Z.Wang,X.Huang and Kai Zhang ,“High-Speed Low-Power Viterbi Decoder Design for TCM Decoders” in IEEE Trans. Very Large Scale Integr.(VLSI) Syst., vol.20,no.4,pp 755-759, April.2012.
- [5] “Bandwidth-efficient modulations,”Consulative Committee For Space Data System,Matera,Italy,CCSDS 401(3.3.6) Green Book, Issue 1,Apr.2003.

BIOGRAPHIES



Valiveti.Ganesh Kumar received his B.E degree in Electronics and Communication Engineering from Saveetha School of Engineering, SAVEETHA UNIVERSITY, Chennai, India in 2012. Currently he is perusing his M.Tech in Vlsi Design in

department of Electronics and Communication Engineering, GITAM Institute of Technology, GITAM UNIVERSITY Visakhapatnam, India.



A.CH.SUDHIR is an Assistant Professor in Department of Electronics and Communication Engineering, GITAM University, Visakhapatnam..Also received PGDES degree in the year 2006.He was awarded M.Tech in