ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

FPGA Implementation of Convolutional Encoder and VD for TCM Decoders Using T-Algorithm

Konangi Naresh Babu
M.Tech (VLSI DESIGN),
Department of ECE,
Pydah Kaushik College of Engineering,
Visakhapatnam, India.

D.P.Raju
Assistant Professor,
Department of ECE,
Pydah Kaushik College of Engineering,
Visakhapatnam, India.

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 TCM decoders. A pre-computation architecture incorporated with T-algorithm for VD, which can effectively reduce the power consumption without degrading the decoding speed much is to be proposed. A general solution to derive the optimal pre- computation steps is also given in the paper.

Keywords: ***Trellis coded modulation (TCM), viterbi decoder, Convolutional code, T-algorithm.***

1. INTRODUCTION

The use of convolutional codes with probabilistic decoding can significantly improve the error performance of a communication system. Trellis coded modulation schemes are used in many bandwidth efficient systems. Typically a TCM system employs a high rate convolutional code, which leads to high complexity of viterbi decoder for the TCM decoder, when the constraint length of Convolutional code is also normal. For example the rate ¾ convolutional code used in trellis coded modulation system for any application has a constraint length of 7 will be in the complexity of the corresponding viterbi decoder for a rate ½ convolutional codes with constraint length of 9 due to the large number of transitions in the trellis. So, In terms of power consumption, the viterbi decoder is dominant module in a TCM decoder. In order to reduce the computational complexity as well as power consumption, low power schemes should be exploited for the VD in a TCM decoder. General solutions for low power viterbi decoder design will be studied in our implementation work. Power reduction in VDs could be achieved by reducing the number of states, (for example reduced state sequence decoding, M-algorithm and T-algorithm) or by over scaling the T-Algorithm has been shown to very efficient in reducing the power consumption. However, searching for the optimal path metric in the feedback loop still reduces the decoding speed. To overcome this drawback, T-Algorithm has proposed in two variations, the relaxed adaptive VD, Which suggests using an estimated optimal path metric, instead of finding the real one each cycle and the limited-search parallel state VD based on scarce state transition [SST]. When applied to high rate convolutional codes, the relaxed adaptive VD suffers a severe degradation of bit-error-rate (BER) performance due to the inherent drifting error between the estimated optimal path metric and the accurate one. On the other hand the SST based scheme requires predecoding and re encoding process and is not suitable for TCM decoders. In TCM, the encoded data are always associated with a complex multi level modulation scheme like 8-ary phase shift keying (8PSK) OR 16/64-ary quadrature amplitude modulation (16/64QAM) through a constellation point mapper. At the receiver, a soft input VD should be employed to guarantee a good coding gain. So, the computational over head and decoding latency due to predecoding and re encoding of the TCM signal become high. An add-compare select unit (ACSU) architecture based on precomputation for VDs incorporating T-Algorithm, which efficiently improves

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

the clock speed of a VD with T-Algorithm for a rate ¾ code. Now, we further analyze the precomputation algorithm. A systematic way to determine the optimal precomputation steps is shown, where the minimum number of steps for critical path to achieve the theoretical iteration bound is calculated and the computational complexity overhead due to precomputation is evaluated. Then, we discuss a complete low-power VD design for the rate ¾ convolutional codes. Finally ASIC implementation results of VD with convolutional encoding are shown.

2. VITERBI DECODER

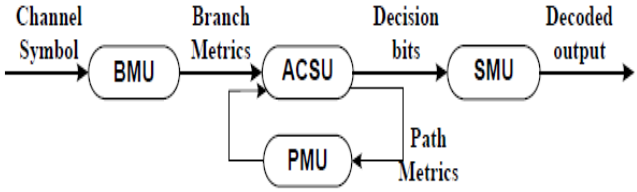

Fig. 1 Viterbi Decoder block diagram

A typical functional block diagram of a Viterbi decoder is shown in Fig. 1. First, branch metrics (BMs) are calculated in the BM unit (BMU) from the received symbols. In a TCM decoder, this module is replaced by transition metrics unit (TMU), which is more complex than the BMU. Then, BMs are fed into the ACSU that recursively computes the PMs and outputs decision bits for each possible state transition. After that, the decision bits are stored in and retrieved from the SMU in order to decode the source bits along the final survivor path. The PMs of the current iteration are stored in the PM unit (PMU).

2.1 Implementation of a Viterbi decoder

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

1. Quantization: Conversion of the analog inputs into digital.

2. Synchronization: Detection of the boundaries of frames and code symbols.

3. Branch metric computation.

4. State metric update: Update the state metric using the new branch metric.

5. Survivor path recording: Tag the surviving path at each node.

6. Output decision generation: Generation of the decoded output sequence based on the survivor path information.

Figure2 shows the flow of the Viterbi decoding algorithm, which performs the above tasks in the specified order.

This section discusses the different parts of the Viterbi decoding process. Analog signals are quantized and converted into digital signals in the quantization block. The synchronization block detects the frame boundaries of code words and symbol boundaries.

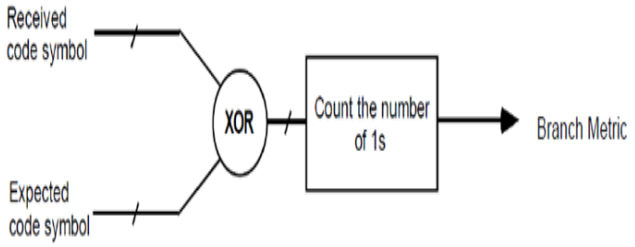

Fig.2 A branch metric computation block

The branch metric computation block compares the received code symbol with the expected code symbol and counts the number of differing bits. An implementation of the block is shown in Figure3

3. PRECOMPUTATION ARCHITECTURE

Precomputation Algorithm

popt(n)=min{p0(n),p1(n),……..p2-1k(n)}=min{min[p0,0(n-1)+B0,0(n),p0,1(n-)+B0,1(n)……., p0,p(n-1)+B0,p(n)], Min[p1,0(n-1)+B1,0(n),p1,1(n-1)+B1,1(n),……,p1,p(n-1) +B1,p(n)], ……., Min[p2k-1-1,0(n-1)+B2k-1-1,0(n),P2k-1-1,1(n-1)+B2k-1-1,1(n),…..,P2k-1-1,p (n-1) +B2k-1-1,p(n)]}=min{P0,0(n-1)+B0,0(n),P0,1(n-1)+B0,1(n),……,P0,p(n-1)+B0,p(n), P1,0(n-1)+B1,0(n),

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

P1,1(n-1)+B1,1(n),….., P1,p(n-1)+B1,p(n),……., P2k-1-1,0(n-1)+B2k-1-1,0(n), P2k-1-1,1(n- 1)+B2k-1-1,1(n),……, P2k-1-1,p(n-1)+B2k-1-1,p(n)}.

Now, we group the states into several clusters to reduce the computational overhead caused by look-ahead computation. The trellis butterflies for a VD usually have a symmetric structure. In other words, the states can be grouped into m clusters, where all the clusters have the same number of states and all the states in the same cluster will be extended by the same Bs. Thus (1) can be rewritten as

Popt = min{min(Ps(n-1)in cluster 1) +min(Bs(n) for cluster 1),Min(Ps(n-1) in cluster 2)+min(Bs(n) for cluster 2), …… , Min(Ps(n-1) in cluster m) +min(Bs(n) for cluster m)}.

The minimum (Bs) for each cluster can be easily obtained from the BMU or TMU and min(Ps) at time n-1 in each cluster can be precalculated at the same time when the ACSU is updating the new Ps for time n. Theoretically, when we continuously decompose Ps(n-1), Ps(n-2),……, the precomputation scheme can be extended to Q steps. Where q is any positive integer that is less than n. Hence Popt(n) can be calculated directly from Ps(n-q) in q cycles.

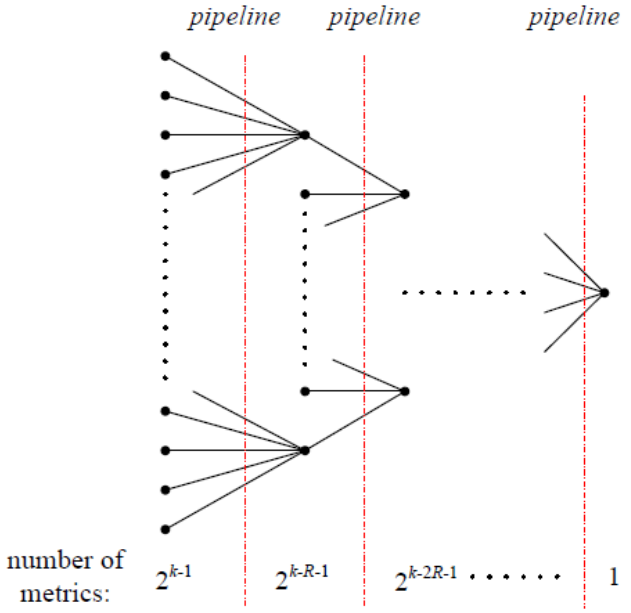

Fig.3 Topology of precomputation pipelining.

Choosing Precomputation steps

In [9], through a design example that, q -step pre-computation can be pipelined into q stages, where the logic delay of each stage is continuously reduced as q increases. As a result, the de- coding speed of the low-power VD is greatly improved. However, after reaching a certain number of steps, qb, further precomputation would not result in additional benefits because of the inherent iteration bound of the ACSU loop. Therefore, it is worth to discuss the optimal number of precomputation steps.

In a TCM system, the convolutional code usually has a coding rate of R/(R+1) , R=2,3,4,……, so that in (1), p=2R and the logic delay of the ACSU is TACSU=Taddder+Tp-in_comp, where Tadder is the logic delay of the adder to compute Ps of each candidate path that reaches the same state and Tp-in_comp is the logic delay of a p-input comparator to determine the survivor path(the path with the minimum metric) for each state. If T-algorithm is employed in the VD, the iteration bound is slightly longer than TACSU because there will be another two input comparator in the loop to compare the new Ps with a threshold value obtained from the optimal Path metric and preset T as shown in (3)

$$\text{Tbound}=\text{Tadder}+\text{Tp\_in\_comp}+\text{T2-in\_comp.} \quad (3)$$

$$q_b = \left\lceil \frac{k-1}{R} \right\rceil \quad (4)$$

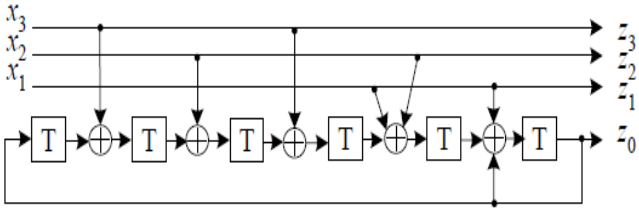

Fig.4 Rate ¾ convolutional encoder.

4. ONE STEP PRECOMPUTATION

For the convenience of our discussion we define the left most register in Fig. 3 as the most significant bit (MSB) and right most register as the least significant

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

bit (LSB). The 64 states and path metrics are labeled from 0 to 63.

A careful study reveals that the 64 states can be partitioned into two groups odd numbered Ps(when „LSB" is 1) And even numbered (when „LSB" is 0) The odd PMs are all extended by odd Bs (when Z0 is „1") and the even PMs are all extended by even Bs (when Z0 is „0"). The minimum P becomes:

Popt (n) = min {min (even Ps (n-1)) + min (even Bs(n)), min (odd Ps (n-1)) +min(odd Bs(n)) }.

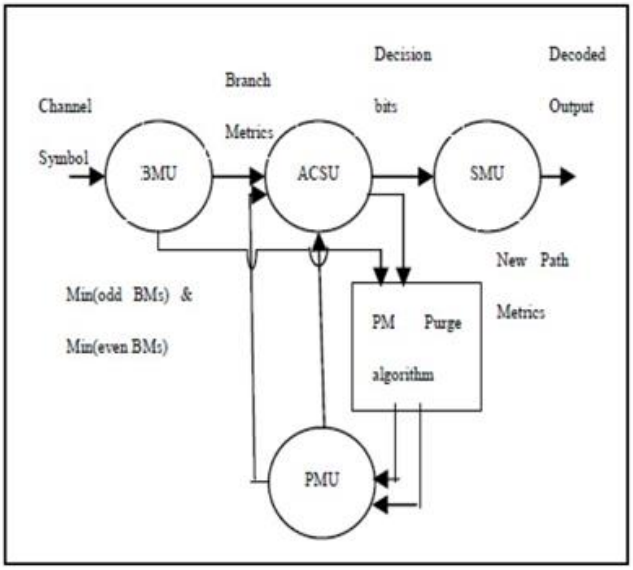

Fig.5 Viterbi decoder with 1-step precomputation T-algorithm

5. TWO STEP PRECOMPUTATION

ACSU Design

We again need to analyze the trellis transition of the original code. In the 1-step pre-computation architecture, we have pointed out that for the particular code shown in Fig. 3, odd-numbered states are extended by odd Bs, while even-numbered states are extended by even Bs. Furthermore, the even states all extend to states with higher indices (the MSB in Fig. 3 is „1") in the trellis transition, while the odd states extend to states with lower indices (the MSB is „0" in Fig. 3). This information allows us to obtain the 2-step pre-computation data path. This process is straightforward, although the mathematical details are tedious. For clarity, we only provide the main conclusion here.

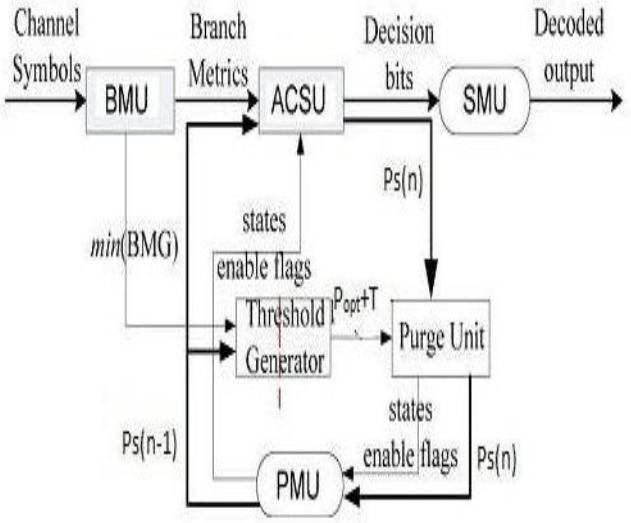

Fig.6 Twostep precomputation T- algorithm

SMU Design

In this section, we address an important issue regarding SMU design when T -algorithm is employed. There are two different types of SMU in the literature: register exchange (RE) and trace back (TB) schemes. In the regular VD without any low-power schemes, SMU always out- puts the decoded data from a fixed state (arbitrarily selected in advance) if RE scheme is used, or traces back the survivor path from the fixed state if TB scheme is used, for low-complexity purpose. For VD in-corporated with T-algorithm, no state is guaranteed to be active at all clock cycles. Thus it is impossible to appoint a fixed state for either out-putting the decoded bit (RE scheme) or starting the trace-back process (TB scheme). In the conventional implementation of T-algorithm, the decoder can use the optimal state (state with Popt), which is always enabled, to output or trace back data. The process of searching for Popt can find out the index of the optimal state as a byproduct. However, when the estimated Popt is used [8], or in our case Popt is calculated from PMs at the previous time slot, it is difficult to find the index of the optimal state.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

6. RESULTS AND DISCUSSIONS

Viterbi decoder with rate ¾ and K=7 is realized by FPGA device xc6slx100l-1Lfgg676. The Synthesis Results for Maximum Clock Speed, Power Estimation Results, device utilization summary, logic utilization and distribution report is shown in Table 6.1. The precomputation VD reduced the power consumption by nearly 70% with minimum decoding speed. The VD design is simulated by Xilinx ISE 14.2.

Convolutional Encoder

The rate ¾ convolutional encoder is being implemented in Xilinx ISE simulator and coded in Verilog HDL.

RTL Schematic

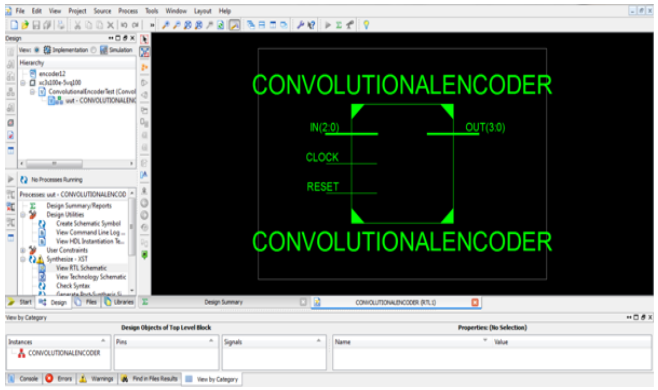

Figure.6.1.RTL Schematic of Convolutional Encoder

Simulation

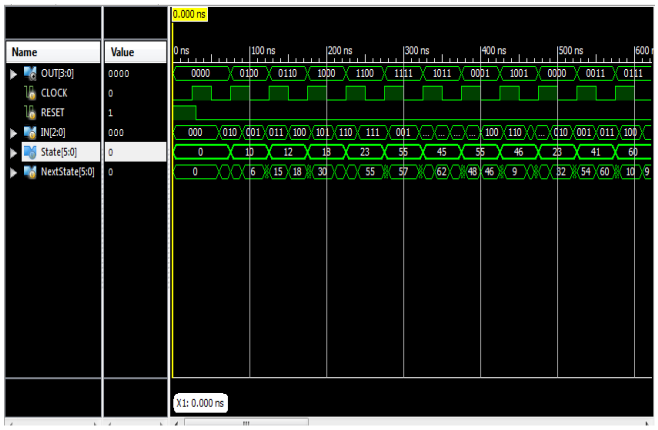

Figure.6.2: Convolutional Encoder with (k/n)=3/4 and K=7.

Synthesis Report

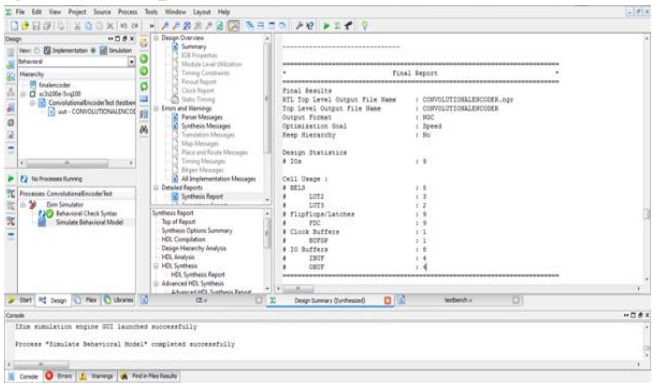

Figure.6.3.Synthesis Report of Convolutional Encoder

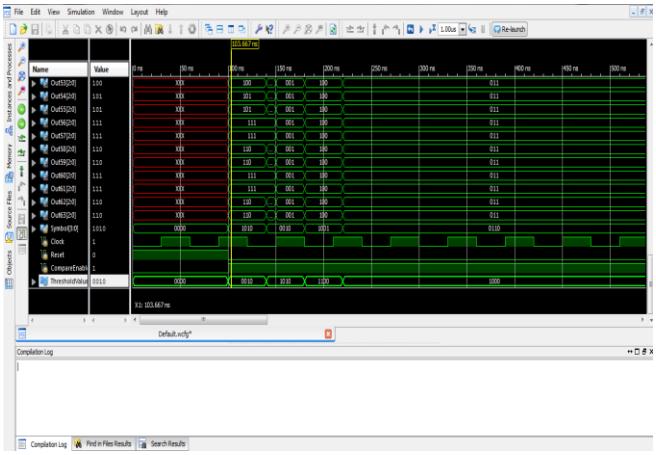

Figure.6.4.Viterbi Decoder with T-algorithm

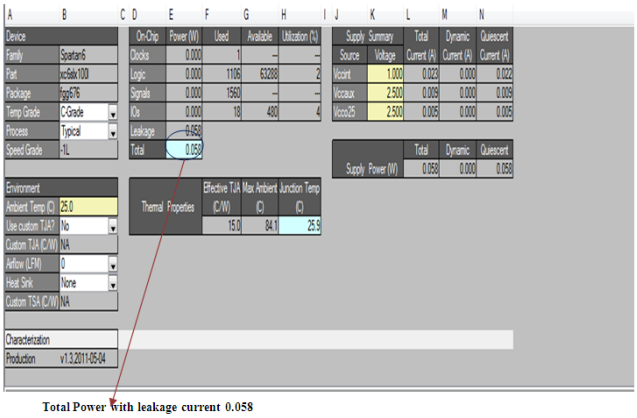

Figure 6.5 Power Report from Xilinx

A. Synthesis Report:

1. Power=60mW
2. Clock Speed

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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 6.1: 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%

Tools Used:

The overviews of tools used in this project are:

Synthesis Tool : Xilinx ISE 14.2
Target Device : xc6slx100l-1Lfgg676
Vendor : Xilinx Corp
Device Family : spartan 6 Low powers
Device : xc6slx100l
Package : Lfgg676
Speed Grade : - 1

7. CONCLUSION

Trellis coded modulation (TCM) schemes are used in many bandwidth-efficient systems. Here a high-speed low-power VD design for TCM systems is proposed. The precomputation architecture that incorporates T–algorithm may reduce the power consumption of VDs without reducing the decoding speed appreciably.

REFERENCES

[1] Jinjin He Sch. of Electr. Eng. & Comput. Sci., Oregon State Univ., Corvallis, OR, USA Huaping Liu ; Zhongfeng Wang ; Xinming Huang ; Kai Zhang "High-Speed Low-Power Viterbi Decoder Design for TCM Decoders" Very Large Scale Integration (VLSI) Systems, IEEE Transactions on Issue 4 April 2012.

[2] Trellis coded modulation" Intutive guide to principles of communicaton www.complextoreal.com

[3]"Trellis coded modulation with redundant signal sets" Gottfride Ungerboeck Feb 1987-Vol. 25,No. 2 IEEE Communications Magazine.

[4] Low power Viterbi decoder for Trellis coded Modulation using T-algorithm" Md.Javeed, B.Sri lakshmi , International Journal of Scientific & Engineering Research International Journal of Scientific & Engineering Research Volume 3, Issue 9, September- 2012 1 ISSN 2229-5518

[5] The viterbi algorithm" A.J. Han Vinck 10.01.2009 Lecture notes data communications Institute for experimental Mathematics Ellernstrasse 29 45326 Essen Germany

[6] Iterative Equalization and Decoding Using Reduced-State Sequence Estimation Based soft-output algorithm" A thesis by Raja Vanktesh Thamma

[7] G Fettweis and H, Meyr, "High-speed parallel Viterbi decoding algorithm and VLSI architecture," IEEE communication Magazine, vol. 29, pp. 46-55, May 1991.

[8] Low power Adaptive Viterbi decoder For TCM Using T-Algorithm" International Journal of Scientific and Research Publications, Volume 3, Issue 8, August 2013 1 ISSN 2250-3153

[9] FPGA Implementation of High Speed And Low Power Viterbi Encoder And Decoder"M.GayathrI, D.Muralidharan M.Tech, School of Computing M.Gayathri et al. / International Journal of Engineering and Technology (IJET).