ISSN 2347-4289

High Speed Viterbi Decoder Design With A Rate Of ½ Convolution Code For Tcm Systems

Anam Srinivasa Reddy, P. Rama Krishna

M. Tech Student, CVSR College of Engineering, Department of Electronics & Communications, A. P. India
2 Associate Professor of Electronics & communication Engineering, Anurag group of Institutions, A.P. India
anam.srinivas@gmail.com, ramakrishnaece@cvsr.ac.in

ABSTRACT: High speed Viterbi decoder design for trellis coded modulation (TCM) is presented in this paper. It is well known that the Viterbi decoder (VD) is the dominant module for determining shortest path. We propose a pre-computation architecture incorporated with T-algorithm for VD, which can find the shortest path 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-1/2 convolution code used in a TCM system shows that compared with the full trellis VD, with the constraint length 9. This work focuses on the realization of convolution encoder and adaptive Viterbi decoder (AVD) with a constraint length (K) of 9 and a code rate (k/n) of ½, Implemented on FPGA. The results are tested by using ISE 10.1 and Modelsim.

Key words: Viterbi decoder, convolution encoder, TCM, T-algorithm, FPGA.

1. INTRODUCTION

The reliability and efficiency of data transmission is the most concerning issue for communication channels in today's digital communications, Error correction technique plays a very important role in communication systems. Convolutional encoding with Viterbi decoding 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 convolution code as they are used in bandwidth-efficient systems. It is highly efficient transmission technique used for transmission of information over band limited channels. This leads to high complexity of the Viterbi decoder (VD) for the TCM decoder, even if the constraint length of the convolution encoder is moderate. Due to the large number of transitions in the trellis diagram power consumption is more in VD. In order to reduce the power consumption as well as the computational complexity, low-power schemes should be exploited for the VD in a TCM decoder. T-algorithm was shown to be very efficient in reducing the power consumption. However, searching for the optimal PM in the feedback loop still reduces the decoding speed. To overcome this drawback, two variations of the T-algorithm have been proposed the relaxed adaptive VD, which suggests using an estimated optimal PM; instead of finding the real one each cycle and the limited-search parallel state VD based on scarce state transition (SST). Because of some drawbacks in both of them, we proposed an add-compare-select unit (ACSU) architecture based on pre-computation incorporating T-algorithm for VDs. This can efficiently improves VDs clock speed.

2. CONVOLUTIONAL ENCODING WITH VITERBI DECODING

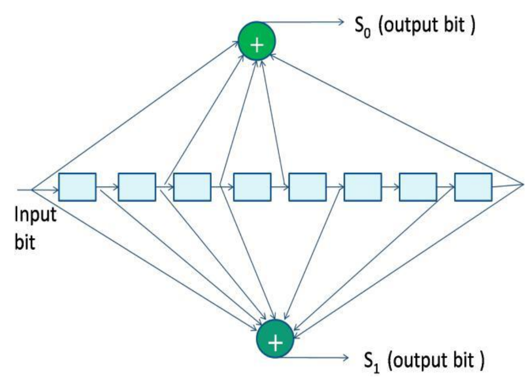

Figure 1: Convolutional encoder with K= 9 and k/n=½

A convolution code is a type of error-correcting code which contains memory and the *n* encoder outputs at any given time unit depend not only on the *k* inputs at that time unit but also on m previous input blocks. Convolution 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 9, equal to the number of stages in the register The encoder has *n* generator polynomials, one for each adder. An input bit is fed into the leftmost register. Using the generator polynomials and the existing values in the remaining registers, the encoder outputs *n* bits. The code rate (k/n) is expressed as a ratio of the number of bits into the Convolutional encoder *k* to the number of channel symbols output by the Convolutional encoder *n* in a given encoder cycle. Convolution encoder with constraint length 9 and code rate ½ is shown in the fig 1. For this encoder we perform the decoding process

ISSN 2347-4289

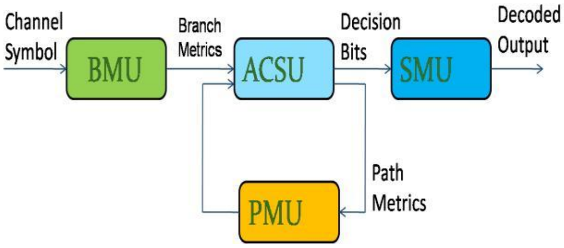

Figure 2: General VD functional diagram

2.1 Viterbi decoder

Viterbi decoder functional block diagram is shown in Fig.2. Branch metrics (BMs) are calculated in the BM unit (BMU) from the received symbols..Then, BMs are fed into the add-compare-select unit (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). ACSU Implementation is critical because the feedback loop makes it the bottleneck for high speed applications. Furthermore, as K value increases, the power consumption and computation complexity increase exponentially. In the T-algorithm, a threshold T is set and the difference between each PM and the optimal one is calculated. So T-algorithm requires extra computation in the ACSU loop for calculating the optimal PM (minimum value of all PMs)and puncturing states.

3. PRE-COMPUTATION ALGORITHM

VD for a convolutional code with a constraint length K Contains 2^{K-1} states and considers each state receives p candidate paths. First, we expand PMs at the current time slot n (PMs(n)) as a function of PMs(n-1) to form a look-ahead computation of the optimal PM - $PM_{opt}(n)$. The Branch metric can be calculated by two types: Hamming distance and Euclidean distance If the branch metrics are calculated based On the Euclidean distance.

$PM_{opt}(n)$ is the minimum value of PMs(n) obtained as

$PM_{opt}(n)$

$= \min \{ PM_0(n) , PM_1(n) ,\ldots\ldots\ldots\ldots PM_{2^{k-1}}(n)$

$= \min \{ \min [PM_{0,0} (n-1)+ BM_{0,0} (n),$
$PM_{0,1} (n-1)+ BM_{0,1} (n) ,\ldots\ldots\ldots\ldots$
$PM_{0,P}(n-1)+ BM_{0,P} (n)] ,$

$\min [PM_{1,0} (n-1)+ BM_{1,0} (n),$
$PM_{1,1} (n-1)+ BM_{1,1} (n) ,\ldots\ldots\ldots\ldots$
$PM_{1,P}(n-1)+ BM_{1,P} (n)] ,$

$\min [PM_{2^{k-1},0} (n-1)+ BM_{2^{k-1},0} (n),$
$PM_{2^{k-1},1} (n-1)+ BM_{2^{k-1},1} (n) \ldots.$
$PM_{2^{k-1},P}(n-1)+ BM_{2^{k-1},P} (n)] \}$..............(1)

For a VD usually the trellis butterflies have a symmetric structure. To reduce the computational overhead caused by look-ahead computation we group the states into several clusters. 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 BMs, The min(BMs) for each cluster can be easily obtained from the BMU or TMU (In a TCM decoder, BMU is replaced by transition metrics unit (TMU), which is more complex than the BMU) and the min(PMs) at time n-1 in each cluster can be pre calculated at the same time when the ACSU is updating the new PMs for time n.The pre computation scheme can be extended to q steps, where q<n (q being any positive integer) Hence, $PM_{opt}(n)$ can be calculated directly from PMs(n-q)in q cycles. The above algorithm (1) when implemented in form of clusters can be rewritten as

$PM_{opt}(n) = \min \{ \min(PMs(n-1) \text{ in cluster } 1)$
$+ \min(BMs(n) \text{ for cluster } 1), \min(PMs(n-1) \text{ in cluster } 2)$
$+ \min(BMs(n) \text{ for cluster } 2),$
……………………

$\min(PMs(n-1) \text{ in cluster } m)$
$+ \min(BMs(n) \text{ for cluster } m) \}$
.... (2)

3.1 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 $T_{ACSU}= T_{adder} +T_{p\text{-}in\text{-}comp}$ .If T-algorithm is employed in the VD, the iteration bound is slightly longer than T_{ACSU} because there will be another two-input comparator in the loop to compare the new PMs with a threshold value obtained from the optimal PM and a preset T and is given by

$$T_{bound}=T_{adder}+T_{p\text{-}in\text{-}comp}+T_{2\text{-}in\text{-}comp} \qquad (3)$$

Where T_{adder} is the logic delay of the adder to compute PMs of each candidate path that reaches the same state and $T_{p\text{-}in\text{-}comp}$ is the logic delay of a p-input comparator (where $p=2^R$) to determine the survivor path for each state.**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 decoding speed of the low-power VD is greatly improved. However, after reaching a certain number of steps, q_b further pre computation would not result in additional benefits because of the inherent iteration bound of the ACSU loop. We limit the comparison to be among only p or 2p metrics, to achieve the iteration bound expressed in (3), for the pre computation in each pipelining stage and assume that each stage reduces the number of the metrics to 1/p (or 2^{-R}) of its input metrics. The smallest number of pre computation steps (q_b) meeting the theoretical iteration bound should satisfy

$$[(2]]^{\uparrow R} \geq 2^{k-1}. \qquad (4)$$

Therefore $q_b \geq \frac{k-1}{R}$ and we express this

as $q_b = \left[\frac{k-1}{R} \right]$

Where [.] denotes ceiling function.

Computational overhead is an important factor that should be carefully evaluated. If there are m remaining metrics after comparison in a stage, the computational overhead from this stage is at least m addition operations. For a code with a constraint length k and q pre computation steps, the number of metrics will reduce at a ratio of $2^{k-1/q}$ and the overall computational overhead is

$N_{overhead.}$ .... (5)

The estimated computational overhead increases exponentially to q. In a real 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 pre computation is a good choice in most cases. For TCM systems, where high-rate convolutional codes are always employed, two steps of pre computation could achieve the iteration bound and also it reduces the computational overhead.

3.2 ACSU Design

Convolutional encoder with Rate ½ and length 9 is shown in fig (1). For convenience of discussion, we define the left-most register in Fig. 1 as the most-significant-bit (MSB) and the right-most register as the least-significant-bit (LSB).The 256 states and PMs are labeled from 0 to 255. The two-step pre-computation in the ACSU feedback loop is expressed as

$PM_{opt}(n)$ = Min[min{
min (cluster 0 (n-2)+ min (BMG0(n-1)),
min (cluster 1 (n-2)+ min (BMG1(n-1)),
min (cluster 2 (n-2)+ min (BMG3 (n-1)),
min (cluster 3 (n-2)+ min (BMG2(n-1)) }
+min(even

BMs(n)), min {
min (cluster 0 (n-2)+ min (BMG1(n-1)),
min (cluster 1 (n-2)+ min (BMG0(n-1)),
min (cluster 2 (n-2)+ min (BMG2 (n-1)),
min (cluster 3 (n-2)+ min (BMG3(n-1)) }
+min(odd BMs(n)) **]**

Where
Cluster 0 = {$PM_m \mid$ 0≤m≤255, m mod 4 =0 };
Cluster 1= { $PM_m \mid$ 0≤m≤255, m mod 4 =2};
Cluster 2= { $PM_m \mid$ 0≤m≤255, m mod 4 =1 };
Cluster 3= { $PM_m \mid$ 0≤m≤255, m mod 4 =3 };
BMG0= { $BM_m \mid$ 0≤m≤15, m mod 4 =0 };
BMG1= { $BM_m \mid$ 0≤m≤15, m mod 4 =2};
BMG2= { $BM_m \mid$ 0≤m≤15, m mod 4 =1 };
BMG3= { $BM_m \mid$ 0≤m≤15, m mod 4 =3};

The functional block diagram of the VD with two-step precomputation T-algorithm is shown in Fig.3 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). The architecture of the TGU is shown in Fig. 4,

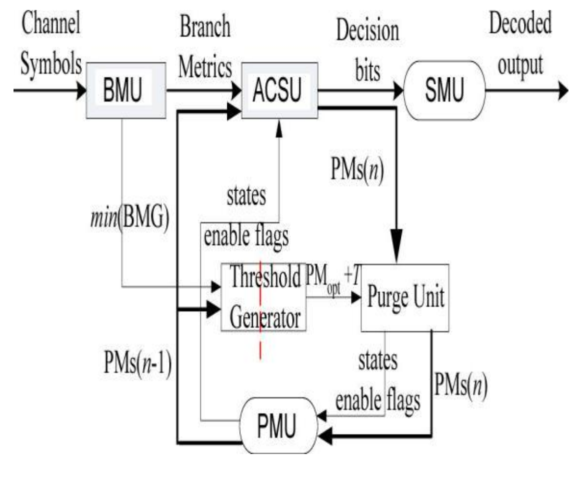

Figure 3: 2-step pre-computation T –algorithm for VD

3.3 SMU Design

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

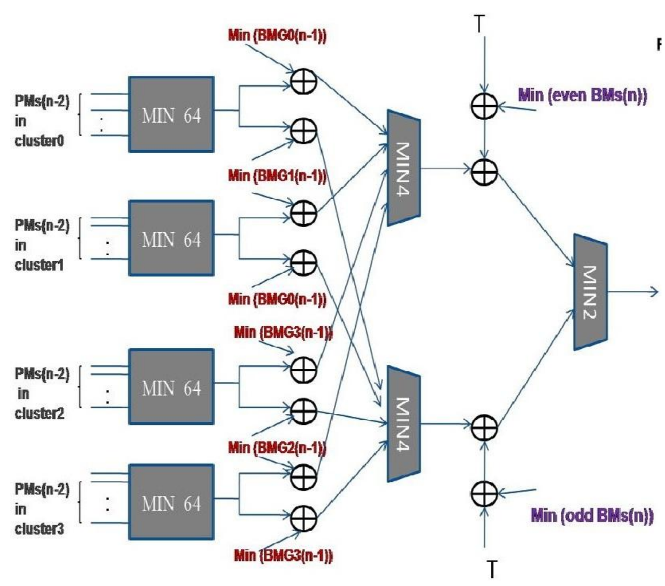

Figure 4: Threshold Generator unit architecture

Thus it is impossible to appoint a fixed state for either outputting 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 PM_{opt}) which is always enabled, to output or trace back data. as the estimated PM_{opt} is calculated from the PMs at the previous time slot. It is difficult to find the index optimal state in the process of searching for the PM_{opt}. A 256 –to- 8 priority encoder can be used for this purpose. The output of the priority encoder would be the unpurged state with the lowest index. Assuming the purged states have the flag “0” and other

ISSN 2347-4289

stages are assigned the flag “1” .implementation of such direct 256-to-8 is not trivial ,so we employ four 4-to-2 priority encoders for the 256 -to -8 priority encoder. This is shown in fig 5.

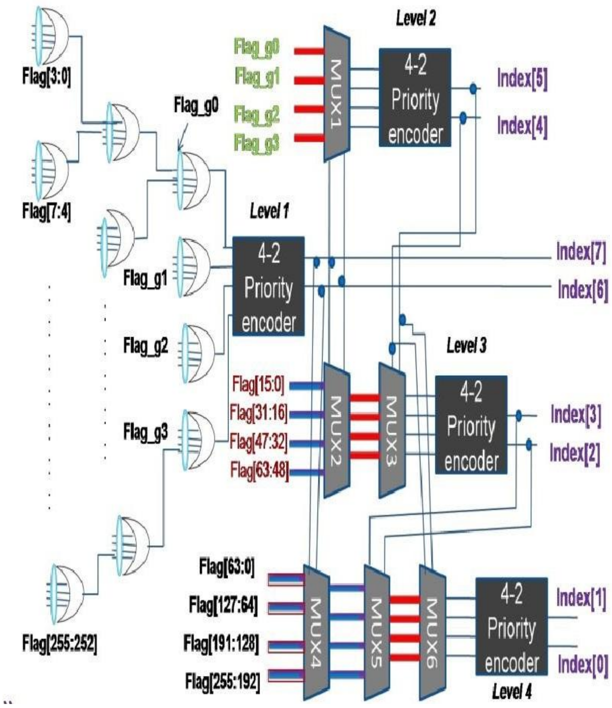

Fig.5: 256-8 priority encoder architecture.

4. SYNTHESIS AND SIMULATION RESULTS

Viterbi decoder with rate ½ and K=9.The device utilization summary, logic utilization and distribution report is shown in Table I. Here the delay also decreasing compared from previous and now that is 69ns to 30ns with minimum decoding speed. The VD design is simulated by Model Sim and Xilinx ISE 10.1.

Table I: Device Utilization Summary (estimated values)

Logic utilization	Used	Avilable	Utilization
No of 4 input LUTs	400	7,168	5%
No of occupied slices	206	3,584	5%
No of bonded IOBs	30	141	21%
No of GCLKs	1	4	25%

5. CONCLUSSION

The pre computation architecture that incorporates T-algorithm efficiently reduces the power consumption of VDs without reducing the decoding speed appreciably. This algorithm is suitable for TCM systems which always employ high-rate convolutional codes. Both the ACSU and SMU are modified to correctly decode the signal. Compared with the full-trellis VD without a low-power scheme, the pre computation VD could has low power consumption with reliable decoding speed. A reusable Viterbi decoder was carried out by adopting the Process Element technique.

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 Integer. (VLSI) Syst., vol. 18, no. 5, pp. 808–817, May 2010.

[3]. 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.

[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.on (VLSI) Systems, Vol. 20, No. 4, April 2012.

[5]. High-Speed Low-Power Viterbi Decoder Design for TCM Decoders Jinjin He,Huaping Liu, Zhongfeng Wang, Xinming Huang, and Kai Zhang IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol. 15, no. 11, pp. 1172– 1176 January 19, 2011.

[6]. 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 2008.

[7]. 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. 2006.

[8]. F. Chan and D. Haccoun, “Adaptive Viterbi decoding of convolutional codes over memory less channels”, IEEE Trans. Commun., vol. 45,no. 11,pp. 1386-1400,Nov2006.

[9]. S. J. Simmons, “Breadth-first trellis decoding with adaptive effort”, IEEE Trans. Commun., vol. 38, no. 1,pp. 3-12, Jan.2004.[6]Bandwidth-efficient modulations, Consultive Committee For Space Data System, Matera, Italy, CCSDS 401(3.3.6)Green Book, Issue 1,Apr. 2003.

ISSN 2347-4289

AUTHORS PROFILES

A. Srinivasa Reddy,
M. TECH (VLSI), CVSR College of Engineering, Ghatkesar, Hyderbad, A.P.

P. RamaKrishna received his B. Tech, M. Tech degree in Electronics and communication engineering (E.C.E), VLSI System Design in the year 2006, 2009 from NIT Warangal, CVR College of engineering JNT University Hyderabad respectively. He had 6 years of teaching and research experience presently he is an Associate professor Anurag group of institutions (autonomous) Hyderabad. His research interests include VLSI System Design, Digital Signal Processing and Image processing.