DEVELOPMENT OF TCM DECODER USING VITERBI DECODER

A.Ravi Teja 1*, Prof. Ch.Srinivasa Kumar 2*

1. II.M.Tech (VLSI), Dept of ECE, AM Reddy Memorial College of Engineering & Technology, Petlurivaripalem.
2. Prof, Dept. of ECE, AM Reddy Memorial College of Engineering & Technology, Petlurivaripalem.

Abstract: *The proposed viterbi decoder plays a dominant role in digital wireless communication for power consumption of the TCM (Trellis Coded Modulation) decoder. This paper gives for power reduction in viterbi decoder achieved by reducing the states. This work gives without degrading the decoding speed we are doing pre-computation architecture with T-algorithm is implemented for this propose, to compare these with full Trellis VD, proposed approach significantly less power consumption. For Low power high speed design of VD for TCM system with optimal time delay is presented in this paper. In this paper work is focus on realization of conventional encoder and Adaptive Viterbi decoder with a constant length and FPGA technology is used. The implemented AVD is analyzed by using ISE10.4 and Modalism simulations.*

1. INTRODUCTION

Viterbi decoding algorithm proposed in 1967 by Viterbi is a decoding process for convolutional codes in memory-less noise. It can be applied to a host of problems encountered in the design of communication systems. Viterbi Algorithm (VA) finds the most-likely state transition sequence in a state diagram given a sequence of symbols. Viterbi algorithm is used to find the most likely noiseless finite-state sequence, it gives a sequence of finite-state signals that are corrupted by noise. In order to reduce the power consumption, increase the speed an asynchronous technique that is Delay Insensitive Null Convention Logic (NCL) for Viterbi Decoder (VD) and its Encoder using Dual rail signal is proposed in this paper. VLSI designs the major cause for the power dissipation is the dynamic power dissipation about 80 to 90 percent of total power dissipation. NCL reduces the dynamic power consumption in terms of reducing the switching activity and also it reduces the Glitch power significantly thereby achieving the lower power. Basically Viterbi algorithm was applied in digital communication systems. It focused on the operations and the practical memory requirement to implement the Viterbi algorithm in real-time. This technique is based on the data generated and decoded from the zero Hamming distance path, and unnecessary computations in the Viterbi decoder was avoided. In that Speed and power were not considered. The Low-power bit-serial Viterbi decoder chip for the code rate r = 1/ 3 and the constraint length K = 9 (256 states) was discussed. Here Add-Compare-Select (ACS) module was based on the bit-serial arithmetic and implemented with the pass transistor logic circuit. And the Scarce State Transition (SST) scheme employed a simple pre-decoder followed by a pre-encoder to reduce the transitions of the Viterbi decoder.

II. RELATED WORK

The convolutional 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. These codes are usually characterized by two parameters code rate (k/n) and constraint length (K) and the patterns of n modulo-2 adders. And shift register has a constraint length (K) of 9 is equal to the number of stages in the register and the encoder has n generator polynomials in that one for each adder. The input bit is fed into the leftmost register. By 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.

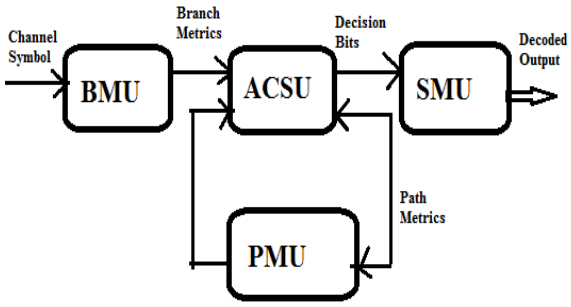

Fig.1. General VD functional diagram.

The Fig.1. Shows Viterbi decoder functional block diagram and Branch metrics (BMs) are calculated in the BM unit (BMU) from the received symbols. BMs are fed into the add-compare-select unit (ACSU) that recursively computes the PMs and outputs decision bits for each possible state transition. Then the decision bits are stored in and retrieved from the SMU in order to decode the source bits along the final survivor path. However PMs of the current iteration are stored in the PM unit (PMU). In the middle 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

**** A.Ravi Teja***
II.M.Tech (VLSI), Dept of ECE, AM Reddy Memorial College of Engineering & Technology,

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.

III. PROPOSED METHODOLOGY

The proposed Viterbe Decoder (VD) for a convolutional code with a constraint length K contains 2K-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 - PMopt(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, PMopt(n) is the minimum value of PMs(n) obtained as

PMopt(n) = min { PM0(n) , PM1(n) ,……. PM2^k-1(n)
= min { min [PM0,0 (n-1)+ BM0,0 (n),
PM0,1 (n-1)+ BM0,1 (n) ,…….
PM0,P(n-1)+ BM0,P (n)] ,
min [PM1,0 (n-1)+ BM1,0 (n),
PM1,1 (n-1)+ BM1,1 (n) ,…….
PM1,P(n-1)+ BM1,P (n)] ,
……
min [PM2^k-1,0 (n-1)+ BM2^k-1,0 (n),

PM2^k-1,1 (n-1)+ BM2^k-1,1 (n) ….
PM2^k-1,P(n-1)+ BM2^k-1,P (n)] } (1)

For Viterbi Decoder (VD) usually the trellis butterflies have a symmetric structure. However to reduce the computational overhead caused by look-ahead computation we group the states into several clusters. 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 PMopt(n) can be calculated directly from PMs(n-q) in q cycles.

Based on above algorithm is implemented in form of clusters can be rewritten as
PMopt(n) = min { min(PMs(n-1) in cluster 1)
+ min(BMs(n) for cluster 1),
min(PMs(n-1) in cluster 2)
+ min(BMs(n) for cluster 2),
min(PMs(n-1) in cluster m)
+ min(BMs(n) for cluster m) } (2)

3.1. PRE-COMPUTATION 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 TACSU= T_{adder} +Tp-in-comp. The 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 PMs with a threshold value obtained from the optimal PM and a preset T and is given by

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

where T_{adder} is the logic delay of the adder to compute PMs of each candidate path that reaches the same state and Tp-in-comp is the logic delay of a p-input comparator (where p=) 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, qb 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 (qb) meeting the theoretical iteration bound should satisfy (2^R)^qb ≥ 2K-1 then

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

The estimated computational overhead increases exponentially to q. And a real design of the overhead increases even faster. Therefore a small number of pre-computational steps are preferred even though the iteration bound may not be fully satisfied. One- or two-step pre-computation is a good choice in most cases. To Design 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.

IV. HIGH SPEED LOW POWER VITERBI DECODER DESIGN

To Design for 4-D 8PSK TCM system with code rate ½, since the pre-computation algorithm always finds the accurate optimal PM, and its BER performance is almost same as that of the conventional T-algorithm.

ACSU DESIGN

For convenience of discussion we define the left-most register as the most-significant-bit (MSB) and the right-most register as the least-significant-bit (LSB). All the 256 states and PMs are labeled from 0 to 255. Here we need these two-step for pre-computation in the ACSU feedback loop is expressed as

PMopt(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 = {PMm 0≤m≤255, m mod 4 =0 };
Cluster 1= { PMm 0≤m≤255, m mod 4 =2};
Cluster 2= { PMm 0≤m≤255, m mod 4 =1 };
Cluster 3= { PMm 0≤m≤255, m mod 4 =3 };
BMG0= { BMm 0≤m≤15, m mod 4 =0 };
BMG1= { BMm 0≤m≤15, m mod 4 =2};
BMG2= { BMm 0≤m≤15, m mod 4 =1 };
BMG3= { BMm 0≤m≤15, m mod 4 =3};

The functional block diagram of the VD with two-step pre-computation T-algorithm is shown in Fig. 2. 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 (PMopt+T). (PMopt+T) and the new PMs are then compared in the "Purge Unit"(PU). The architecture of the TGU is shown in Fig. 3,

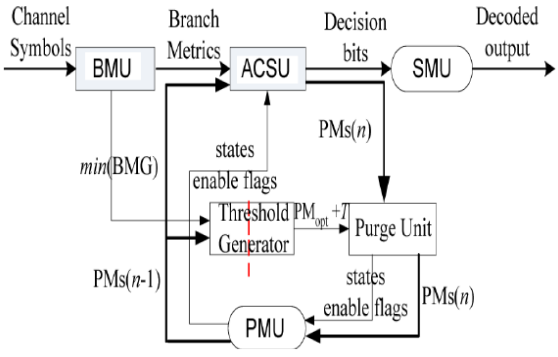

Fig.2. Two Step Pre-computation T-algorithm for VD.

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. 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 PMopt) which is always enabled, to output or trace back data. As the estimated PMopt is calculated from PMs at the previous time slot, it is difficult to find the index of the optimal state in the process of searching for the PMopt . 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 states 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. 3. and it is simpler also.

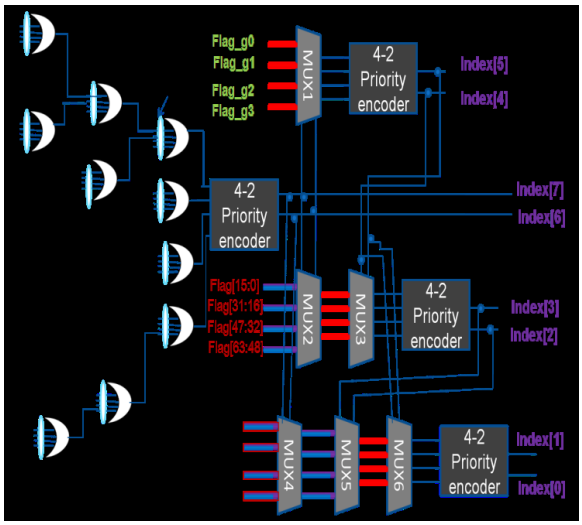

Fig.3. 256-to-8 priority encoder architecture.

PROCESS ELEMENT TECHNIQUE:

Different protocols use different convolutional code and varied applications have different requirement for throughput, area and power. So design of reusable Viterbi decoder is important, too. In present project, a reusable Viterbi decoder was carried out. This decoder adopted the Process Element (PE) technique, which made it easy to adjust the throughput of the decoder by increasing or decreasing the number of PE. By the method of Same Address Write Back (SAWB), we reduced the number of registers to half in contrast with the method of ping-pong.

This decoder supported punctured convolution code and was data-driven, which means the circuit was able to work under different data rate and avoid those invalid operations. The core parameters, such as the generation words of convolution code,

the number of PE, the depth of TBU and maybe the radix of butterfly, are all configurable. By assembling different numbers of PE, we can get the state-serial, part parallel or full parallel structure of Viterbi decoder. And because the PMU is scattered into each PE, this structure is more area efficient.

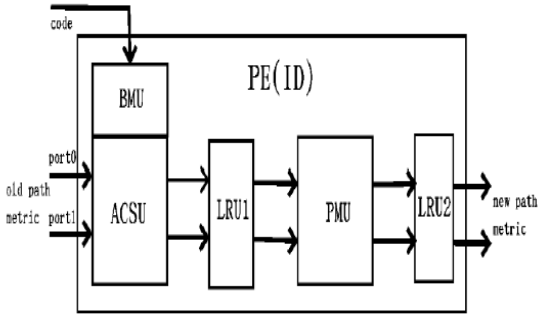

Fig.4. Structure of process element.

V. CONCLUSION

The proposed pre-computation designed architecture incorporate T-algorithm efficiently reduces the power consumption of VDs without reducing the decoding speed appreciably. For High speed Low power consumption this algorithm is suitable for TCM system with always employ high rate convolutional codes. In this article both ACSU and SMU are modified to correctly decode the signal. The pre-computation VD could has low power consumption with reliable decoding speed.

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, 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] Joshi M.V., Gosavi S., Jegadeesan V., Basu A., Jaiswal S., Al-Assadi W.K. and Smith S.C. 2007, NCL Implementation of Dual-Rail 2s Complement 8×8 Booth2 Multiplier using Static and Semi-Static Primitives, IEEE region 5 Technical Conference, April 20-21, Fayetteville,59- 64.

[6] Jun Jin Kong, Keshhab K Parhi., 2004 Low- Latency Architectures for High-Throughput Rate Viterbi Decoder, IEEE Transactions on VLSI System, 12(6), 642-651.

[7] Meilana Siswanto1, Masuri Othman, Edmond Zahedi,2006 VLSI Implementation of 1/2 Viterbi Decoder for IEEE P802.15-3a UWB Communication, IEEE ICSE2006 Proc., Kuala Lumpur, Malaysia,666 – 670.

[8] Qing Li, Xuan-zhong Li, Han-hong Jiang and Wen-hao He 2008, A High-Speed Viterbi Decoder, Fourth International Conference on Natural Computation IEEE.,p.p. 313-316.