ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

Speed and Power Optimization of FPGA'S Based on Modified Viterbi Decoder

V.Prasanth, Rubia Tasneem

Assoc. Professor, Department of Electronics & Communications, Pragati Engineering College, A.P, India.

M. Tech Student, Department of Electronics & Communications, Pragati Engineering College, A.P, India.

ABSTRACT: Advancement in the VlSI technology using low power, less area and high speed constraints is mostly used for encoding and decoding of data. In this paper power and cost reduction with increase in speed using viterbi decoder(VD) for trellis coded modulation(TCM) is proposed. Viterbi decoder uses viterbi algorithm for TCM decoding, but the efficient speed and power reduction is not achieved at the receiving ends. A pipelined architecture with a pre-computational approach which incorporated T-algorithm for VD is proposed in this paper. Priority encoder is used along with convolution encoders to send data bits of high priority first with a code rate of ½ in TCM system. The proposed architecture reduces power consumption of 80% without performance loss. The degradation of clock speed used in the architecture is negligible. Proposed architecture is simulated and synthesized using Xilinx ISE successfully and analyzed using FPGA Spartan-6 XC6SLX45 which is a low power target device. The results obtained are found to be consuming low power.

KEY WORDS: Priority encoder, Spartan-6 XC6SLX45, Trellis coded modulation (TCM), Viterbi decoder.

I. INTRODUCTION

The decoding process for convolution codes in memory less noise systems is done by using viterbi algorithm proposed by viterbi. For the host problem encountered in the communication systems this algorithm can be applied. Many band efficient systems employ Trellis coded modulation technique[1].A high-rate convolution code, which leads to a high complexity of the viterbi decoder (VD) for the TCM decoder, even if the constraint length of the convolution code is moderate is employed in TCM systems. For example, the rate -3/4 convolution code used in a 4-D TCM system for deep space communications [2] has a constraint length of 7; however, the computational complexity of the corresponding VD is equivalent to that of a VD for a rate-1/2 convolution code with a constraint length 9 due to the large number of transitions' in the trellis.

The viterbi decoder in TCM decoders is dominant module in terms of power consumption. To reduce the computational complexity as well as power consumption, low-power schemes should be exploited for the VD in a TCM decoder. The General solutions for low-power VD designs have been well studied by existing work. power reduction in VDs could be achieved by reducing the number of states for example, reduced-state sequence decoding(RSSD)[3].M-algorithm[4] and T-algorithm[5] or[6] by over scaling the supply voltage[7].The whole system that includes the VD should be taken into consideration for over-scaling the supply voltage, which is not main focus of our research. M-algorithm in practical applications is efficient algorithm compared to RSSD[3],because the M-algorithm requires a sorting process in a feedback loop while T-algorithm only searches for the optimal path metric(PM),that is the minimum value or the maximum value of all PMs.

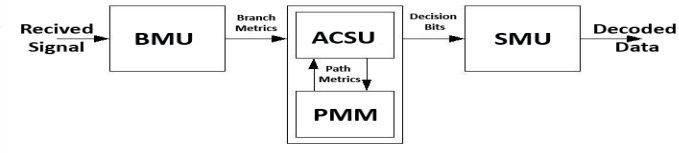


Fig.1. Viterbi decoder functional diagram.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

For reducing the power consumption T-algorithm has been shown to be very efficient. [8], [9]. However, searching for the optimal PM in the feedback loop still reduces the decoding speed which is a drawback. To overcome this two variations of the T -algorithm have been proposed: the relaxed adaptive VD [8], 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) [9]. In our preliminary work [10], we have shown that when convolution codes of high rate are applied the relaxed adaptive VD suffers a severe degradation of bit-error-rate (BER) due to which performance inherent drifting error occurs between the estimated optimal PM and the accurate one. On the other hand, the SST based scheme requires pre-decoding and re-encoding processes and is not suitable for TCM decoders. 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 in TCM scheme. A soft-input VD should be employed at the receiver to guarantee a good coding gain. Due to which pre-decoding and re-encoding of the TCM signal along with the computational overhead and decoding latency become high. In our preliminary work [10], based on pre-computation, an add-compare-select unit (ACSU) architecture for VDs incorporating T –algorithm was proposed, which efficiently improves the clock speed of a VD with T -algorithm for a rate-3/4 code. In this work, we further analyze the pre-computation algorithm. A systematic way to determine the optimal pre-computation 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 pre-computation is evaluated. Then, we discuss a complete low-power high-speed VD design for the rate-3/4 convolution code [2]. Finally low power fpga implementation results of the VD are reported, which have not been obtained in our previous work in [10].

The remainder of this paper is organized as follows. Section II Gives the background information of VDs. Section III presents the pre-computation architecture with T-algorithm and also discusses the choice of pre- computation steps. Details of a design example including the modification of survivor-path memory unit (SMU) are discussed in Section III. Synthesis and power estimation results are reported in Section IV and conclusions are given in Section V. T - algorithm and discusses the choice of pre- computation steps. Details of a design example including the modification of survivor-path memory unit (SMU) are discussed in Section III. Synthesis and power estimation results are reported in Section IV and conclusions are given in Section V.

II.VITERBI DECODER

A typical Viterbi decoder functional block diagram is shown in Fig. 1. First, in the BM unit BMU branch metrics (BMs) are calculated from the received symbols. This module is replaced by transition metrics unit (TMU) in a TCM decoder, which is more complex than the BMU .Then, BMs are fed into the ACSU that recursively which computes the PMs and outputs decision bits for each possible state transition. After that, the decision bits are stored in they are retrieved from the SMU in order to decode the source bits along the final survivor path. The PM unit (PMU) stores the PMs of the current iteration. An extra computation in the ACSU loop for calculating the optimal PM and puncturing states is required in T -algorithm. Due to which, a straight- forward implementation of T-algorithm will dramatically reduce the decoding speed ,The key point of improving the clock speed of T -algorithm is to quickly find the optimal PM.

III. PRECOMPUTATION ARCHITECTURE

A. Pre-computation Algorithm

The pre-computation algorithm idea was presented in [9]. Consider a convolution code for VD with a constraint length of k, where each state receives p candidate paths. First, we expand PMs at the current time slot n(PMs(n)) as a function of n(PMs(n-1)) to form a look-ahead computation of the optimal PM-PM opt(n). If the branch metrics are calculated based on the Euclidean distance,

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

$$PM_{opt}(n) = Min\{PM_0(n),\ PM_1(n), \ldots\ldots PM_{2-1}^{K}(n)\} = Min\{Min[\,PM_{0,0}(n-1) + BM_{0,0}(n),\ PM_{0,1}(n-1) + BM_{0,1}(n), \ldots\ldots PM_{0,P}(n-1) + BM_{0,P}(n)],\ min[PM_{1,0}(n-1) + BM_{1,0}(n),$$

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

$$PM_{1,1}(n-1) + BM_{1,1}(n), \ldots\ldots$$
$$PM_{1,P}(n-1) + BM_{1,P}(n)],$$
$$\ldots\ldots\ldots\ldots\ldots\ldots,$$
$$min[PM_2^{K-1}{}_{-1,0}(n-1) + BM_2^{K-1}{}_{-1,0}(n),$$
$$PM_2^{K-1}{}_{-1,1}(n-1) + BM_2^{K-1}{}_{-1,1}(n)$$
$$PM_2^{K-1}{}_{-1,P}(n-1) + BM_2^{K-1}{}_{-1,P}(n)]\}$$
$$= min\{PM_{0,0}(n-1) + BM_{0,0}(n),$$
$$PM_{0,1}(n-1) + BM_{0,1}(n), \ldots\ldots\ldots,$$
$$PM_{0,P}(n-1) + BM_{0,p}(n),$$
$$PM_{1,0}(n-1) + BM_{1,0}(n),$$
$$PM_{1,1}(n-1) + BM_{1,1}(n), \ldots\ldots\ldots,$$
$$PM_{1,P}(n-1) + BM_{1,P}(n),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,$$
$$PM_2^{K-1}{}_{-1,0}(n-1) + BM_2^{K-1}{}_{-1,0}(n),$$
$$PM_2^{K-1}{}_{-1,1}(n-1) + BM_2^{K-1}{}_{-1,1}(n),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,$$
$$PM_2^{K-1}{}_{-1,P}(n-1) + BM_2^{K-1}{}_{-1,P}(n)\}$$

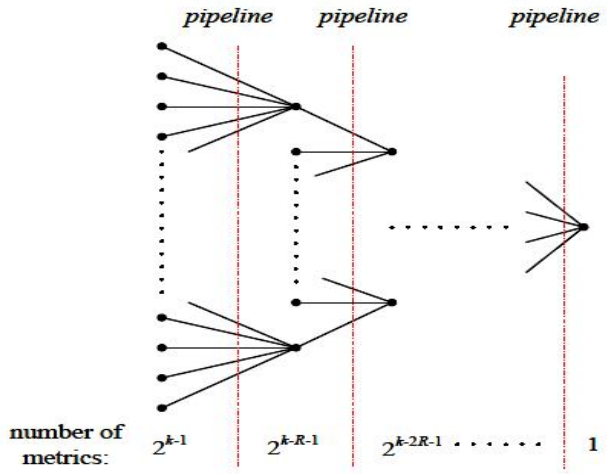


Fig. 2. Topology of pre-computation pipelining

Then, the states are grouped into several clusters in order to decrease the computational overhead caused by look-ahead computation. Usually the trellis butterflies have a symmetric structure for VD. In other words, the states can be grouped into 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.

Thus, (1) can be rewritten as:

$PM_{opt}(n)$ = Min { Min (PMS (n – 1) in cluster1)
Min (BMS (n) for cluster1),
Min (PMS (n – 1) in cluster2)
Min (BMS (n) in cluster2),
……..,
Min (PMS (n – 1) in cluster m)
Min (BMS (n) in cluster m) }

For each cluster the min(BMs) can be easily obtained from the BMU or TMU and the min(PMs) at time n -1 in each

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

cluster at the same time they can be pre-calculated when the ACSU is updating the new PMs for time n. Theoretically, when we continuously decompose PMs(n - 1); PMs(n - 2); . .. . ., the pre-computation scheme can be extended to q steps, where q is any positive integer that is less than n.Hence, PMopt(n) can be calculated directly from PMs(n-q) in q cycles

B. Selection of Pre-computation steps

In [9], we have shown through a design example that, q-step pre-computation can be pipelined into q stages, As q increases the logic delay of each stage is continuously reduced. 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 ASCU loop.Therefore it is worth to discuss the optimal number of pre-computation steps. In TCM system, the convolution code usually has a coding rate of R/R+1,R=2,3,4,.....so that in (1),$p=2^R$ and the logic delay of the ACSU is $T_{ACSU}=T_{adder}+T_{p\text{-}in\text{-}comp}$ 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 to determine the survivor path(the path with minimum metric) for each state., the iteration bound is slightly longer than T_{ACSU} because there will be another two-input comparator in the loop to compare the new PM and a preset T as shown below if T-algorithm is employed in the VD (3)

$$T_{bound}=T_{adder}+T_{p\text{-}in\text{-}comp}+T_{2\text{-}in\text{-}comp}$$

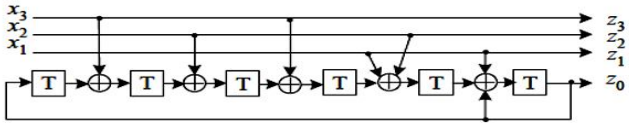


Fig.3. Rate ¾ convolution encoder.

In each pipelining stage to achieve the iteration bound expressed in(3),for the pre-computation, we limit the comparison to be among only p or 2p metrics. To simplify our evaluation ,we assume that each stage reduces the number of metrics to 1/p(or2^{-R}) of its input metric as shown in fig.2.The smallest number of pre computation steps(q_b) meeting the theoretical iteration bound should satisfy $(2R)q_b \geq 2k-1$.Therefore,$q_b \geq (k-1)/R$ and q_b is expressed as (4),where $\lceil . \rceil$ denotes the ceiling function

$$q_b = \frac{\lceil k-1 \rceil}{R}$$

In design example shown in [9], with coding rate of ½ and constraint length of 7, the minimum pre computation steps for VD to meet the iteration bound is 2 according to (4).It is the same value as we obtained from direct architecture design[9].In some cases, the number of remaining metrics may slightly expand during a certain pipeline stage after addition with BMs. The extra delay can be absorbed by an optimized architecture or circuit design usually,. Even it is hard to eliminate the extra delay, the resultant clock speed is much closer to the theoretical bound. we could add another pipeline stage, to fully achieve the iteration bound, though it is very costly as will be shown next. Computational overhead (compared with conventional T -algorithm) is an important factor that should be carefully evaluated. The most computational overhead comes from adding BMs to the metrics at each stage as indicated in (2). In other words, if there are m remaining metrics after comparison in a stage, the computational overhead from this stage is at least m addition operations. The exact overhead varies from case to case based on the convolution code's trellis diagram. Again, we consider a code with a constraint length k and q pre-computation steps to simplify the evaluation. Also, we assume that each remaining metric would cause a computational overhead of one addition operation. In this case, the number of metrics will reduce at a ratio of $2^{(k-1)/q}$ and the overall computational overhead is (measured with addition operation).

$$N_{overhead} = 2^0+2^{(k-1)/q}+2^{(k-1)/q}+.....+2^{(q-1)(k-1)/q}$$

$$= \frac{2^{q(k-1)/q}-1}{2^{(k-1)/q}-1} \quad = \frac{2^{(k-1)}-1}{2^{(k-1)/q}-1}$$

The estimated computational overhead accordin Min(BMG) $2^{6/q}$-1) when k = 7 and q ≤ 6, which increases almost exponentially to q . In a real design, the overhead ... faster than what is given by (5) when other factors (such as comparisons or expansion of metrics as we mentioned above) are taken into consideration. Therefore, a small number of pre-computational steps is preferred even though the iteration bound may not be fully satisfied. The good choice in most cases one or two-step pre-computation. The above analysis also reveals that pre-computation is not a good option for low-rate convolution codes (rate of 1/Rc; R_c = 2; 3; . . .), because it usually needs more than two steps

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

to effectively reduce the critical path (in that case, R = 1 in (4) and qb is k - 1). The two steps of pre-computation could achieve the iteration bound or make a big difference in terms of clock speed where high-rate convolution codes are employed for TCM systems. Due to which, the computational overhead is small.

IV. LOW-POWER HIGH-SPEED VITERBI DECODER DESIGN

We are still using the 4-D8PSK TCM system as described in [2] as example. The rate-3/4 convolution code employed in the TCM system is shown in Fig.3.Preliminary BER performance and architecture design for the ACSU unit have been discussed in [10].In section, we further address the SMU design issue the code rate ½ is employed. Later in the next section, we report various FPGA family implementation results that have not been obtained before.

BER performance of the VD employing T-algorithm with different values of T over an additive white Gaussian noise channel is shown in Fig.4.The simulation is based on a 4-D 8PSK TCM system employing the rate3/4 code

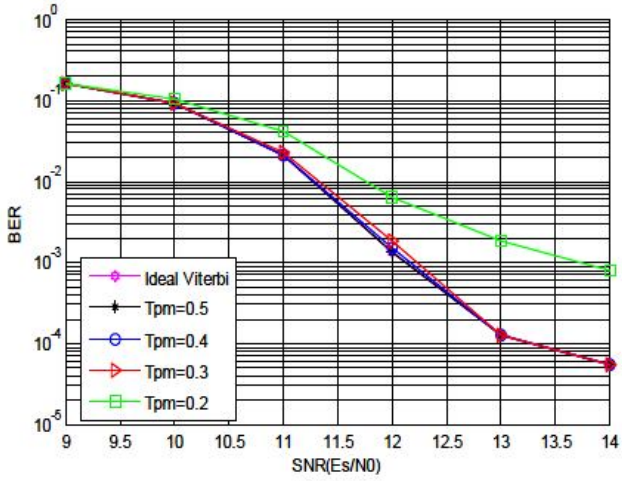


Fig.4. BER Performance of T-algorithm

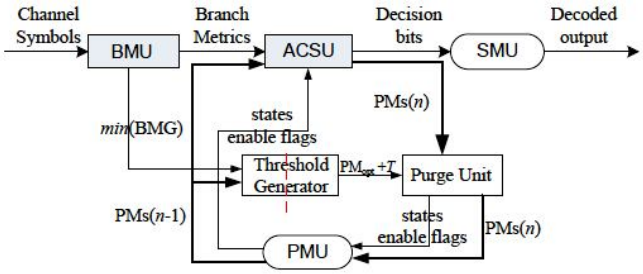


Fig.5. VD with two-step pre-computation T-algorithm

The overall coding rate is 11/12 after due to other unencoded bits in TCM system. Compared with ideal viterbi algorithm, the threshold "T pm"can be lowered to 0.3 with less than 0.1db of performance loss; while the computational complexity could be reduced by upto 90% ideally .Since the performance is the same as that of convectional T-algorithm

A.ACSU Design

We have concluded that two-step pre-computation is the optimal choice for the rate-3/4 code VD. For convenience of discussion, we define the left-most register in Fig.3as the most significant bit(MSB) and the rightmost register as the least significant-bit(LSB).The 64 states and PMs are labeled from 0 to 63.The two-step pre-computation is expressed as

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

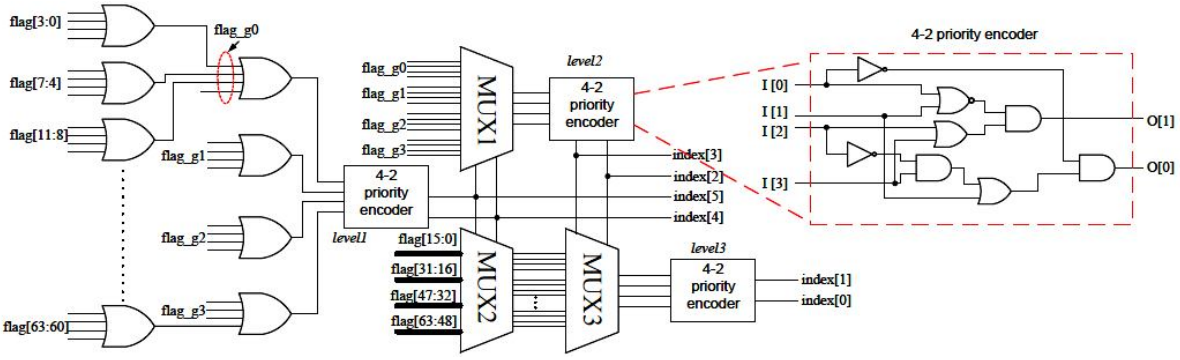


Fig.6. Architecture of 64-to-6 priority encoder

$$PM_{opt}\ (n) = PM_{opt}\ (n)$$
$$Min\ [\ Min\ \{\ Min\ cluster\ 0\ (n-2) + Min\ \big(BMG0\ (n-1)\big),$$
$$Min\ (cluster\ 1(n-2) + Min\ \big(BMG1\ (n-1)\big),$$
$$Min\ (\ cluster\ 2\ (n-2) + Min\ \big(BMG3\ (n-1)\big),$$
$$Min\ (cluster\ 3\ (n-2) + Min\ \big(BMG2\ (n-1)\big)\}$$
$$+ \text{Min (even BMS (n))},$$
$$Min\ [\ Min\ \{\ Min\ cluster\ 0\ (n-2) + Min\ \big(BMG0\ (n-1)\big)$$
$$\text{Min (Cluster 1 (n} - 2)) + \text{Min (BMG0 (n} - 1)),$$
$$\text{Min (Cluster 2 (n} - 2)) + \text{Min (BMG2 (n} - 1)),$$
$$\text{Min (Cluster 3 (n} - 2)) + \text{Min (BMG3 (n} - 1))\}$$
$$\text{Min (Odd BMS (n))}]$$

Cluster0={ $PM_m|0 \le m \le 63, m \bmod 4=0\}$;
Cluster1={ $PM_m|0 \le m \le 63, m \bmod 4=2\}$;
Cluster2={ $PM_m|0 \le m \le 63, m \bmod 4=1\}$;
Cluster3={ $PM_m|0 \le m \le 63, m \bmod 4=3\}$;
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 VD with two-step pre-computation T-algorithm functional block diagram is shown in fig.5. The minimum value of each BM group(BMG) is calculated in BMU or TMU and then passed to the "Threshold Generator "unit (TGU) to calculate (PM_{opt}+T). The new PMs and (PM_{opt}+T) are then compared in the "purge unit"(PU).The architecture of the TUG is shown in Fig6 in, which the key functions of the two-step pre-computation scheme are implemented. In Fig.6,the "MIN 16" unit for finding the minimum value in each cluster is constructed with two stages of four-input comparators. To meet the iteration bound this architecture has been optimized, compared with the convectional T-algorithm ,the computational overhead of this architecture is 12 addition operations and a comparison, which is slightly more than the number obtained from the evaluation(5).

B. SMU Design

An important issues regarding SMU design when T-algorithm is employed in this section is addressed. Two different types of SMU in the literature: register exchange (RE) and the trace back (TB) schemes are there. The SMU always outputs the decoded data from a fixed state in the regular VD without any low power schemes (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-corporate with T-algorithm, no state is guaranteed to be active at all clock cycles.
Thus it is not possible to appoint a fixed state for either outputting the decoded bit (RE scheme) or starting the trace-back process(TB scheme).In the implementation of convectional T-algorithm, the optimal state (state with PM_{opt}) can

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

be used by the decoder, which is always enabled, to output or trace back data. The process of searching for PMopt can find out the index of the optimal state as a byproduct. However, when the estimated PMs at the previous time slot, it is difficult to find the index of the optimal state. A practical method is to find index of an enabled state through a $2^{(k-1)}$ to –(k-1) priority encoder. Suppose that we have labeled the states from 0 to 63.The output of the priority encoder would be the unpurged state with the lowest index. Assuming the other states are assigned the flg"1" and purged states have the flag"0",the truth table of such priority encoder is shown in table1,where the "flag" is the input and "index" is the output

Input (I[3:0])	Output(O[1:0])
x x x 1	0 0
x x 0 1	0 1
x 1 0 0	1 0
1 0 0 0	1 1

C. Threshold Generator Unit

To generate the threshold value a threshold generator unit is used which will take maximum or minimum value at each stage. It is not trivial to implement such a table. We employ an efficient architecture for 64-to-6 priority encoder based on three 4-to-2 priority encoders in our design, as shown in the below Fig 7.The 64 flags each of which contains 16 flags are first divided into 4 groups,. The priority encoder which is at level 1 detects which group contains at least one"1" and determines index[5:4]".Then MUX2 selects one group of flags based on "index[5:4]".The input of the priority encoder which is at level2 can be computed from output of MUX2 by "or" operations. By introducing another MUX (MUX1) we can reuse the intermediate results. The output of the priority encoder which is at level 2 is "index[3:2]".Again "index[3:2]" selects four flags (MUX3) as input of the priority encoder level 3. Finally, the last encoder will determine "index [1:0]".Implementing the 4-to-2 priority encoder is much simpler than implementing the 64-to-6 priority encoder. Its truth table is shown in Table II and corresponding logics are shown in (7) and (8)

$$O[0] = \overline{I[0]} \cdot (I[1] + I[3] \cdot \overline{I[2]} \cdot \overline{I[1]})$$
$$= \overline{I[0]} \cdot (I[1] + I[3] \cdot \overline{I[2]});$$
$$O[1] = \overline{I[0]} \cdot \overline{I[1]} \cdot (I[2] + \overline{I[2]}\, I[3])$$
$$= \overline{I[0] + I[1]} \cdot (I[2] + I[3])$$

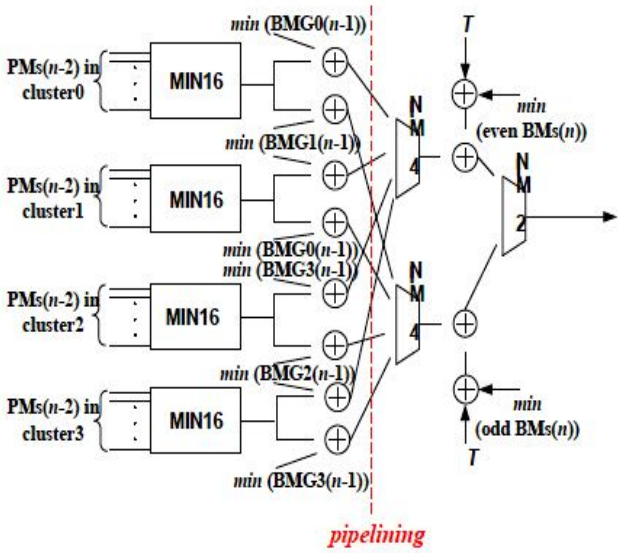


Fig.6. Architecture of TUG

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

	Max speed(MHz)	Cell area(mm^2)
Full-trellis VD	505	0.58
VD with 2-step pre-computation	446.4(-11.6%)	0.68(+17.2%)
Conventional T-algorithm	232(-54.1%)	0.685(+18%)

TABLE 3: Synthesis Results for Maximum Clock Speed

V. IMPLEMENTATION RESULTS

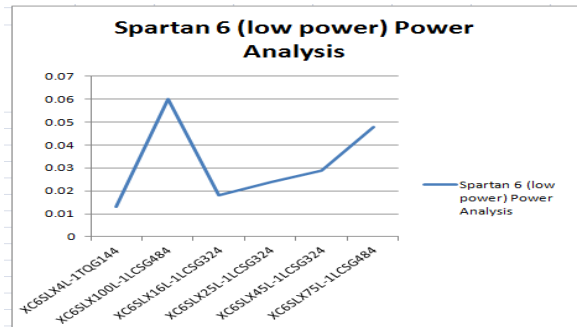


Fig.7.Spartan 6 low power analysis

Power Analysis	
FPGA families	**Spartan 6 low power Analysis**
XC6SLX4L-1TQG144	0.013
XC6SLX100L-1LCSG484	0.06
XC6SLX16L-1LCSG324	0.018
XC6SLX25L-1LCSG324	0.024
XC6SLX45L-1LCSG324	0.029
XC6SLX75L-1LCSG484	0.048

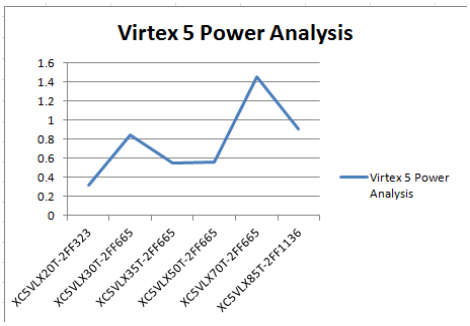


Fig.8. Virtex 5 power Analysis

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

Table 5: Virtex 5 families

FPGA families	Basic viterbi decoder power analysis	Viterbi decoder with T-algorithm
XC6SLX25T-2CSG324	0.051	0.031
XC6SLX45-2CSG324	0.059	0.039
XC6SLX75-2CSG484	0.084	0.067
XC6SLX4-2TQG144	0.036	0.016
XC6SLX75T-2CSG484	0.087	0.012
XC6SLX100-2CSG484	0.105	0.083

TABLE 6: Power Analysis of Spartan 6

Power Analysis	
FPGA families	**Virtex 5 families**
XC5VLX20T-2FF323	0.323
XC5VLX30T-2FF665	0.843
XC5VLX35T-2FF665	0.549
XC5VLX50T-2FF665	0.565
XC5VLX70T-2FF665	1.454
XC5VLX85T-2FF1136	0.909

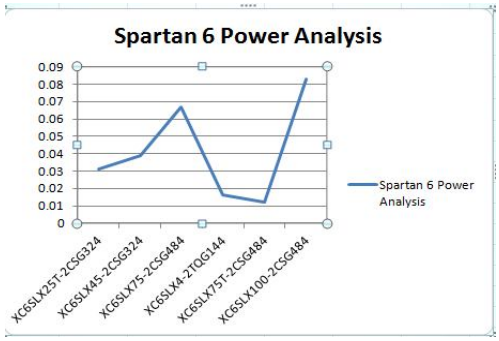


Fig .9.Spartan 6 families

TABLE 7: Power comparison for Spartan 6 families

Power Analysis	
FPGA families	**Spartan 6 power Analysis**
XC6SLX25T-2CSG324	0.031
XC6SLX45-2CSG324	0.039
XC6SLX75-2CSG484	0.067
XC6SLX4-2TQG144	0.016
XC6SLX75T-2CSG484	0.012
XC6SLX100-2CSG484	0.083

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

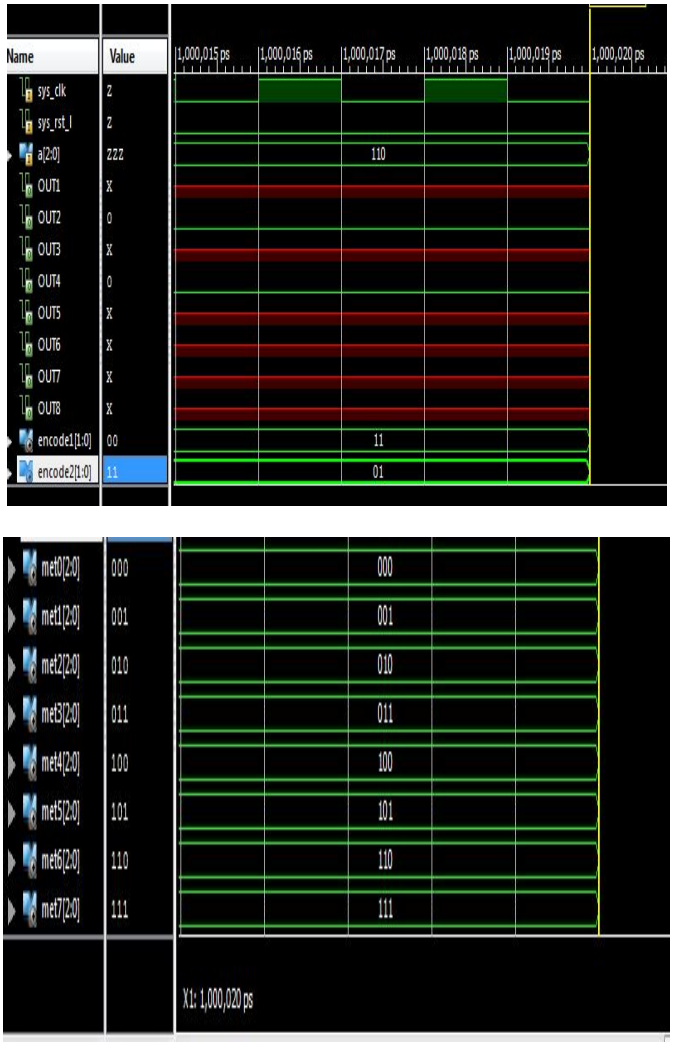
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

Table 8: Power comparision of Virtex 5 families

FPGA families	Basic viterbi decoder power analysis	Viterbi decoder with T-algorithm
XC5VLX20T-2FF323	0.345	0.323
XC5VLX30T-2FF665	0.866	0.843
XC5VLX35T-2FF665	0.571	0.549
XC5VLX50T-2FF665	0.587	0.565
XC5VLX70T-2FF665	1.48	1.454
XC5VLX85T-2FF1136	0.932	0.909

Simulation Result



The full-trellis VD, the VD with the two-step pre-computation architecture and one with the conventional T-algorithm

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

are modeled with Verilog HDL code.The soft inputs of all VDs are quantized with 7 bits.Each PM in all VDs is quantized as 12 bits.RE scheme with survival length of 42 is used for SMU and the register arrays associated with the purged states are clock-gated to reduce the power consumption in SMU.For ASIC synthesis, We use TSMC 90-nm CMOS standard cell. The synthesis targets to achieve the maximum clock speed for each case and the result are shown in table III

Table II shows that the VD with two-step pre-computation architecture only decreases clock speed by 11% compared with the full-trellis VD. Mean while, the increase of the hardware area is about 17%.Thee decrease of clock speed is inevitable since the iteration bound for VD with T-algorithm is inherently longer than that of full-trellis VD. Also, any kinds of low-power scheme would introduce extra hardware like purge unit shown in Fig5 or the clock-gating module in the SMU. Therfore, the hardware overhead of the proposed VD is expected. On the other hand, the VD with conventional T-algorithm cannot achieve half of the clock speed of the full trellis VD.

Therefore, for high-speed applications, it should not be considered. It is worth to mention that the conventional T-algorithm VD takes slightly more hardware than proposed architecture, which is counterintuitive. This is because the former decoder has much longer critical path and the synthesis tool took extra measures to improve clock speed. It is clear that the conventional T-algorithm is not suitable for high-speed applications. If the target throughput is moderately high proposed architecture can operate at a lower supply voltage

which is counterintuitive. This is because the former decoder has much longer critical path and the synthesis tool took extra measures to improve the clock speed. It is clear that the conventional T-algorithm is not suitable for high speed application. If the target throughput is moderately high, the proposed architecture can operate at lower supply voltage, which will lead to quadratic power reduction compared to the conventional scheme. Thus we next focus on power comparison between the full trellis VD and the proposed scheme.

We estimate the power consumption of these two designs with synopsis prime Power under the clock speed of 200Mb/s(power supply of 1.0 V,temperature of 300K).A total of 1133 received symbols(12000 bits) are simulated. The results are shown in table IV.

With the finite word-length implementation, the threshold can only be changed by a step of 0.125.Therfore, to maintain a good BER performance, the minimum threshold we chose is 0.375.Table IV shows that, as the threshold decreases, the power consumption of the proposed VD is reduced accordingly. In order to achieve the same BER performance, the proposed VD only consumes 30.8% the power of the full-trellis VD.

VI. CONCLUSION

We have proposed a high-speed low power VD design for TCM systems. The pre-computation architecture that incorporates T-algorithm efficiently reduces the power consumption of VDs without reducing the decoding speed appreciably. We have also analyzed the pre-computation algorithm, where the optimal pre-computation steps are calculated and discussed. The algorithm is suitable for TCM systems which always employ high-rate convolution codes. Finally, we presented a design case. Both the ACSU and SMU are modified to correctly decode the signal.ASCI synthesis and power estimation results show that, compared with full-trellis VD without a low-power scheme, the pre-computation VD could reduce the power consumption by 80%with only 11% reduction of the maximum decoding speed.

REFERENCES

[1] High-Speed Low-Power Viterbi Decoder Design for TCM Decoders Jinjin He, Huaping Liu, Zhongfeng Wang, Xinming Huang, and Kai Zhang
[2] "Bandwidth-efficient modulations," Consultative Committee For Space Data System, Matera, Italy, CCSDS 401(3.3.6) Green Book, Issue 1, Apr. 2003.
[3] 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 1994.
[4] C. F. Lin and J. B. Anderson, " -algorithm decoding of channel con- volutional codes," presented at the Princeton Conf. Info. Sci. Syst., Princeton, NJ, Mar. 1986.
[5] S. J. Simmons, "Breadth-first trellis decoding with adaptive effort,"*IEEE Trans. Commun.*, vol. 38, no. 1, pp. 3–12, Jan. 1990.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2014

[6] F. Chan and D. Haccoun, "Adaptive viterbi decoding of convolutional codes over memoryless channels," *IEEE Trans.*
[7] R. A. Abdallah and N. R. Shanbhag, "Error-resilient low-power viterbi decoder architectures," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4906–4917, Dec. 2009.
[8] 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.
[9] F. Sun and T. Zhang, "Low power state-parallel relaxed adaptive viterbi decoder design and implementation," in *Proc. IEEE ISCAS*, May 2006, pp. 4811–4814.
[10] J. He, H. Liu, and Z. Wang, "A fast ACSU architecture for viterbi de- coder using T-algorithm," in *Proc. 43rd IEEE Asilomar Conf. Signals, Syst. Comput.*, Nov. 2009, pp. 231–235.
[11] 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
[12] Arkadiy Morgenshtein, M Moreinis and "Asynchronous Gate- Diffusion-Input (GDI) Transactions on VLSI Systems, 12, pp.847-856.
[13] El-Dib D. A. and M. I. Elmasry.2004, "Modified Viterbi decoder for low power wireless communCircuits Syst. I, Reg. 51, pp. 371–378.
[14] Song li and qing-ming yi., 2006. 'The Design of Power Consumption Bidirectional Viterbi Decoder Conference on Machine Learning and Cybernetics
[15] Mohammad K.Akbari and Ali Jahanian, 2004, Power and Robust design for Add Compare Proceedings of the IEEE Conferecne on EURO Digital System Design (DSD '04).
[16] Arkadiy Morgenshtein, Fish, and I. A. Wagner input (GDI) – A novel power efficient method detailed methodology" in Proc. 14th IEEE Int. 39–43.
[17] Arkadiy Morgenshtein, M Moreinis and "Asynchronous Gate-Diffusion-Input (GDI) Transactions on VLSI Systems, 12, pp.847-856.
[18] F. Chan and D. Haccoun. Adaptive Viterbi Decoder Codes over Memoryless Channels. IEEE Communications, 45(11):1389–1400, Nov. 1997.
[19] http://www.xilinx.com/products/silicon-devices/fp
[20] Denton J. Daily (2004), "Programming LogicXilinx ISE and CPLDs," in Prentice Hall, 203 pager Communication and Informatics (ICCCI - 2013), Jan. 04 – 06