An Efficient Low Power Viterbi Decoder Design using T-algorithm

K. Sridhar Reddy
M. Tech, Student
Department of ECE
Vardhaman College of Engineering

M. Nagarjuna
Assistant Professor
Department of ECE
Vardhaman College of Engineering

H. Shravan Kumar
Assistant Professor
Department of ECE
Vardhaman college of Engineering

ABSTRACT

This paper presents an efficient Low-Power Viterbi Decoder Design using T-algorithm. It implements the viterbi decoder using T-algorithm for decoding a bit-stream encoded by a corresponding forward error correction convolutional encoding system. A lot of digital communication systems incorporated a viterbi decoder for decoding convolutionally encoded data. The viterbi decoder is able to correct errors in received data caused by channel noise. We proposed an architecture implementing a Viterbi Decoder with T-algorithm deployed with threshold generator unit and purge unit to reduce the number of states which reduce power consumption. We propose modified architecture for the survivor Metric Unit to reduce the memory Access power during the trace back operation. The proposed viterbi decoder is carried out for rate-1/2 with a standard constraint length 7. The Synthesis results will be done using cadence RTL Encounter Tool. For ASIC synthesis, we use TSMC 45-nm CMOS Process. The architecture which reduces the complexity and power Consumption by as much as 70% without effecting the decoding speed.

General Terms

Low-Power, Viterbi Decoder

Keywords

Viterbi Decoder (VD), Convolutional Code, Constraint Length, Code rate, VLSI, Trellis Diagram

1. INTRODUCTION

Now-a-days Convolutional Codes are today one of the mostly used technique for correcting error's in modern digital Wireless Communication systems. Convolutional encoder and Viterbi decoder are deployed in modern digital communication systems as a part of forward error correction technique mechanism [2]. As the convolution codes are used mostly for the channel encoding of data to achieve low-error rate in latest wireless communication standards like 3GPP, GSM and WLAN. All communication channels are subject to the additive white Gaussian noise (AWGN). Convolutional codes are coded the incoming input bit stream continuously serial manner. In convolutional codes, the block of encoded digits generated by encoder in a time unit depends not only on the block of K message digits within that time unit but also on the preceding n-1 blocks of message bits. These Viterbi decoders are used in high speed wireless data transmission where data rates upto the Mega-bits per second a convolution code vector is generated by combining the outputs of a K-state shift register through the employment of EX-OR logic summer

The Block Diagram of Viterbi Decoder in digital communications is shown in Fig: 1

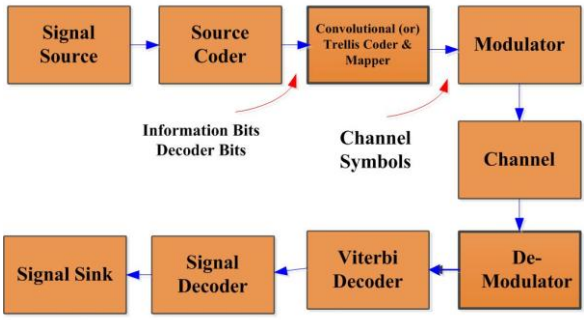

Fig 1: Viterbi Decoder in Digital Communication System.

The convolution encoder will encode the input bit stream continuously in serial manner. The encoder will adds some redundant bits it generates a code-vector. The code-vector will be transmitted through a channel at a data rate upto the hundreds of Mega-bits per second. The Viterbi decoder will be decode the encoded data as a Original input bit stream. We proposed the convolutional encoder with a rate R=1/2 and standard constraint-length (L) =7 to reduce the complexity as well as power consumption [1]. The Top Module of a Viterbi Decoder is shown in Fig: 2.

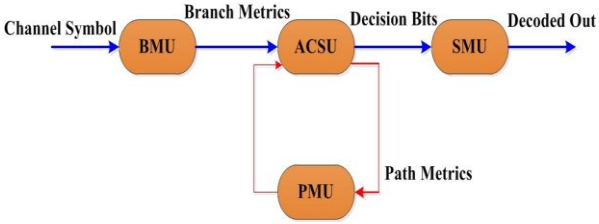

Fig 2: Top Module of Viterbi Decoder

In this paper we proposed an efficient architecture for the add-compare-select-unit with purge unit and Threshold Generator unit (TGU), So that the complexity and power will be reduced in the Viterbi decoder using power reduction technique. VD's achieved power reduction by reducing the number states using T-algorithm [4]. The T-algorithm contains the pre computation steps will be involved. The minimum number of steps for the critical path is to be calculated. Finally the ASIC Implementation results of the VD are reported.

2. VITERBI DECODER

The Architecture of a viterbi decoder is shown in Fig: 3. the soft-decision bits are fed into Branch Metric Unit (BMU's). The BMU calculates Branch Metrics (BM's) from the received input bits. The BMs are given as input to the ACSU which continuously computes the PM's. The TGU [1] will find the minimum path based on the Branch Metrics and the new path metric will find in the purge unit. The Output decision bits are stored in Memory Unit. The minimum path is retrieved from the SMU by using Trace-back Method in to decode the input bits of the Convolutional encoder along the final survivor path. The pre-computation steps of the iterations are stored in the PM Unit.

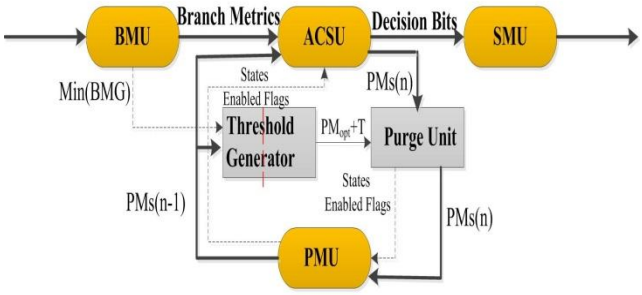

Fig 3: Architecture of Viterbi Decoder

T-algorithm has pre-computation in the ACS loop in-order to calculate the path metrics and puncturing states. The T-algorithm is used to find the optimal path by reducing the no of states.

3. T-ALGORITHM

The T-algorithm has pre-computation steps. In T-algorithm the survivor paths are not constant. For every survivor path the trellis stage l-1 is expanded and its success value at stage l are kept if their corresponding path metric values are smaller or equal to d_m +T. where d_m is the minimum value and T is the Threshold value determined by the user. The T-algorithm [4] reduces the number of states so the complexity of the computation decreases. The T-algorithm [4] is the sorting process or comparison operation for searching the best path or minimum path metric in the decoding stage.

3.1 Pre-Computation Method

The Pre-computation Algorithm is shown below. Consider a VD for a Convolutional code with a constraint length k, where each state receives P candidate paths. If the branch metrics are calculated based on the Euclidean distance [3], the optimal PM becomes the minimum value of all the PMs. The minimum Pre-computation steps are shown below.

PM_{opt} (M)

= min {PM_0 (n), PM_1 (n) ……. $PM^k{}_{2-1}$(n)}

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

$PM_{0,1}$ (n-1) + $BM_{0,1}$(n)……

$PM_{0,p}$ (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)……

$PM_{1,p}$ (n-1) + $BM_{1,p}$(n)],

……..

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),

$PM_{0,p}$ (n-1) + $BM_{0,p}$(n),

$PM_{1,0}$ (n-1) + $BM_{1,0}$(n),

$PM_{1,1}$ (n-1) + $BM_{1,1}$(n),

$PM_{1,p}$ (n-1) + $BM_{1,p}$(n),

……

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)}

Then we divided the above computation into several clusters in order to reduce the complexity of the computations. The trellis butterflies [5] usually have a symmetric structure for a VD. Here the states can be grouped into m clusters, where all the clusters have the same number of states. Then the same BMs is rewritten has

PM_{opt} (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)}

The min (BMs) can be obtained from 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. Theoretically, when we continuously decompose the pre-computation scheme can be 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, PM_{opt} can be calculated directly from PMs(n-q) in q cycles. The Topology of pre-computation pipelining to find number of metrics is shown in Fig: 4.

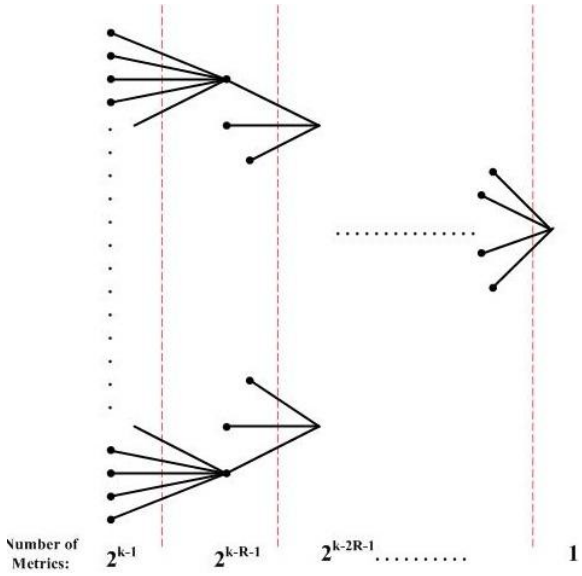

Fig 4: Topology of pipelining Architecture

4. PROPOSED ARCHITECTURE

I. Convolutional Encoder Design

The Convolutional Encoder Design that is carried out with Rate R=1/2 and Constraint Length (L) =7. For 2 bit Soft decision VD. The Convolutional Encoder with 1 input bit and 2 output bits $S_1$and S_2. The block Diagram of Convolutional is shown in Fig: 5.

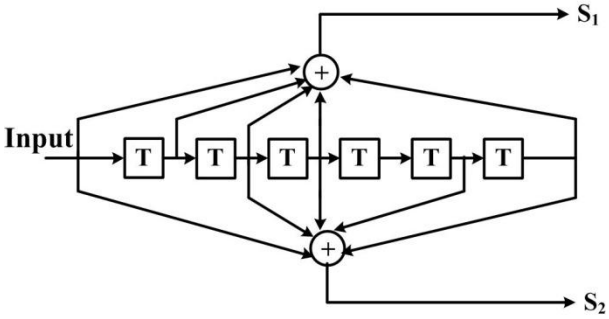

Fig 5: Convolutional Encoder with R=1/2 and L=7

A Convolutional encoder adds some redundant bits into the data stream using the Linear-Shift Register. The Message bits are input into shift register and the encoded output bits are obtained by Modulo-2 addition of the input Message bits and the shift register and the encoder will produce 2 bits of encoded information for each input bit information. The Truth Table of Convolutional encoder is shown in Table 1

Table 1. Truth table of convolutional encoder

Input	S1	S2
1	1	0
0	1	0
1	0	0
1	1	1

II. Viterbi Decoder Design

A. Branch Metric Unit

The BMU which calculates the Branch Metrics by soft decision input bits. The Branch Metric Calculator (BMC) computes the Branch Metrics Using Euclidean distances [9]. The r received symbol and faded symbol are used to calculate the branch metrics.

Branch Metric Calculator

The Branch Metric directly computes the Euclidean distance and the calculated distances will be stored it in memory. The branch Metrics are simply read. The Branch Metric Calculator [6] is shown in Fig: 6

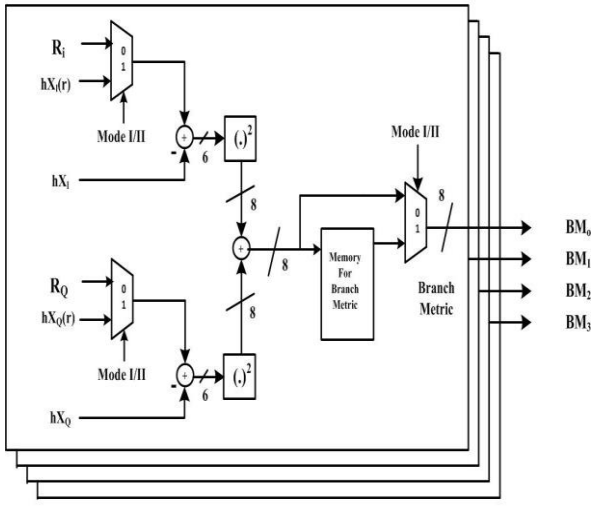

Fig 6: Branch Metric Calculators

B. Add and compare select unit

In the ACSU design. BM's are accumulated in the Path Metric Unit (PMU) to determine the decoding path in the trellis diagram. We proposed an architecture using T-algorithm. The T-algorithm uses the Pre-computation steps. The ACSU [10] reads BM from the memory passed to the Threshold Generator Unit to calculate the Computation $\{PM_{opt} + T\}$ and the comparator will compares the path metrics and the best path will be stored in memory as a decision bits. The Purge Unit will calculate the new path using computation steps. The ACS Unit is shown in Fig: 7.

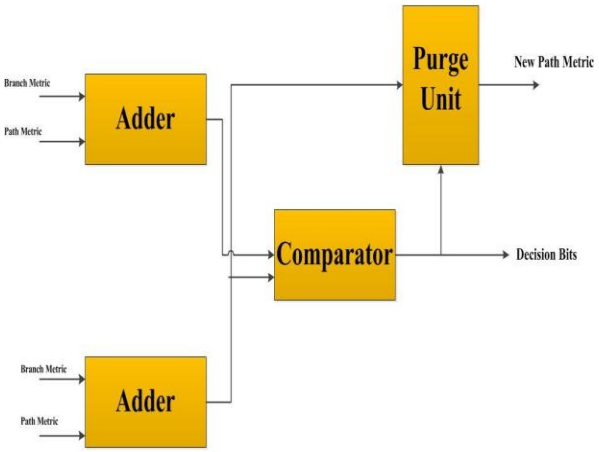

Fig 7: Add and compare select unit

Threshold Generator Unit

The Threshold Generator Unit architecture is shown which calculates the pre-computation steps. In TGU will use pipelining structure to find pre-computation steps the MIN 16 will find Minimum value in each Cluster group with two stages of four input Comparators using pipelining structures. The Topology of pipelining [1] structure is shown in Fig: 4. the architecture will calculate the pre-computation steps and reduces the no of states. This decreases the power consumption. The Threshold Generator is the modified architecture of ACSU. The architecture of TGU is shown in Fig: 8.

C. Survivor Metric Unit

In this section, when we employed T-algorithm in VD then there will be two issues will be raised. There are two different types of SMU presented in this paper: trace-back (TB) method and Register Exchange (RE) [6] method. In the regular VD, if RE is used SMU always outputs the decoded data from a fixed state. If TB is used it will trace back [8] the survivor path from the fixed state. Here we proposed Modified TB method. The decoder will use the optimal state PM_{opt} which is enabled always and trace back the output. In this SMU a practical method to find the index of enabled state is 2^{k-1} to (k - 1) priority encoder. The architecture of 64-to-6 priority encoder is labeled the states from 0 to 63. This is an efficient architecture based on three 4-2 priority encoder as shown in Fig: 9. there are the 3 Levels in the architecture which sub-press the 64 bit to 6 bits. Implementing the 4-to-2 priority encoder is used to implement the 64-to-4 priority encoder. The architecture of 64 –to-6 priority encoder is shown Fig: 10.

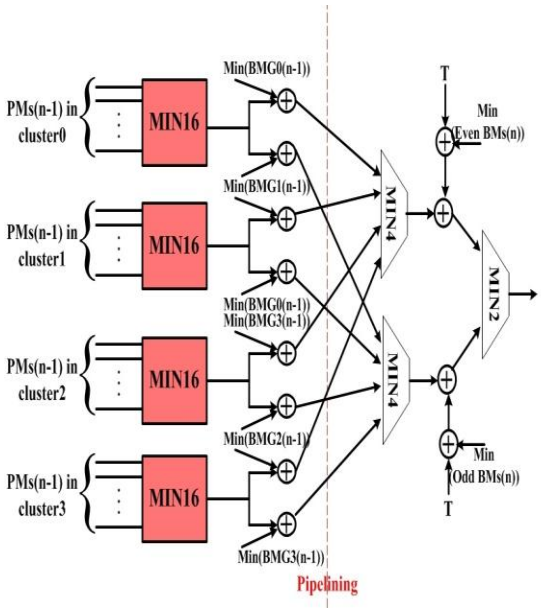

Fig 8: Architecture of TGU

5. IMPLEMENTATION RESULTS

The efficient Low-Power Viterbi Decoder Design is implemented using Cadence Tool. The Implementation has been done using Verilog HDL Code. The synthesis Results has been done using Cadence RTL Encounter Tool. The synthesis Results of Encoder is shown in Fig: 10 and the synthesis Result of Decoder is shown in Fig: 11. The Power Calculations has been done using the cadence RTL Encounter tool. The Power Report is shown in Table II. For ASIC, We use TSMC 45 nm CMOS Standard Cell. The ASIC Synthesis has been done using the Cadence Encounter RTL to GDSII Tool. The CMOS Standard cell Using TSMC 45 nm is shown in Fig: 12.

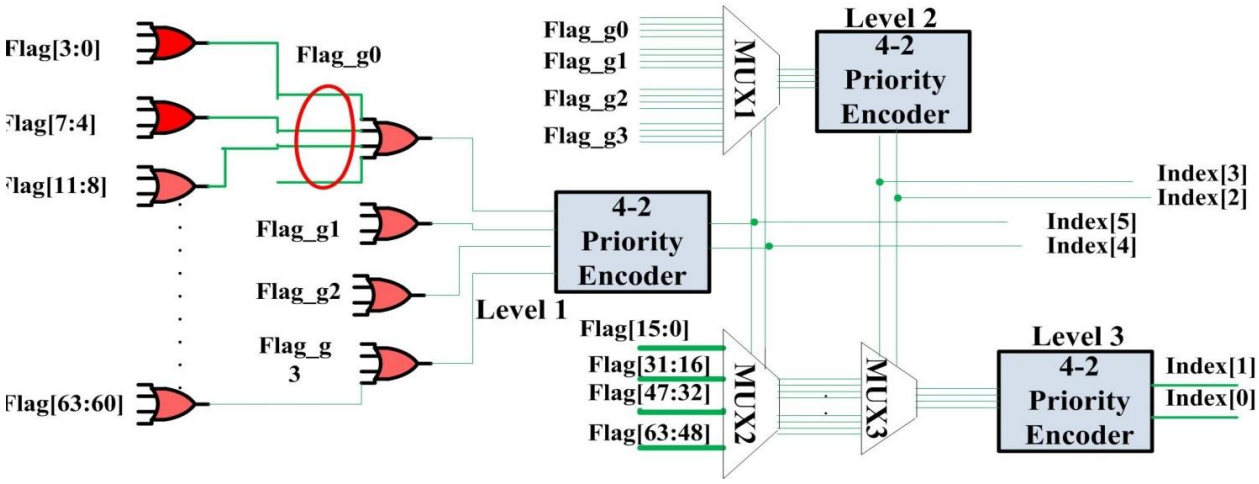

Fig 9: Architecture of 64-to-6 priority encoder

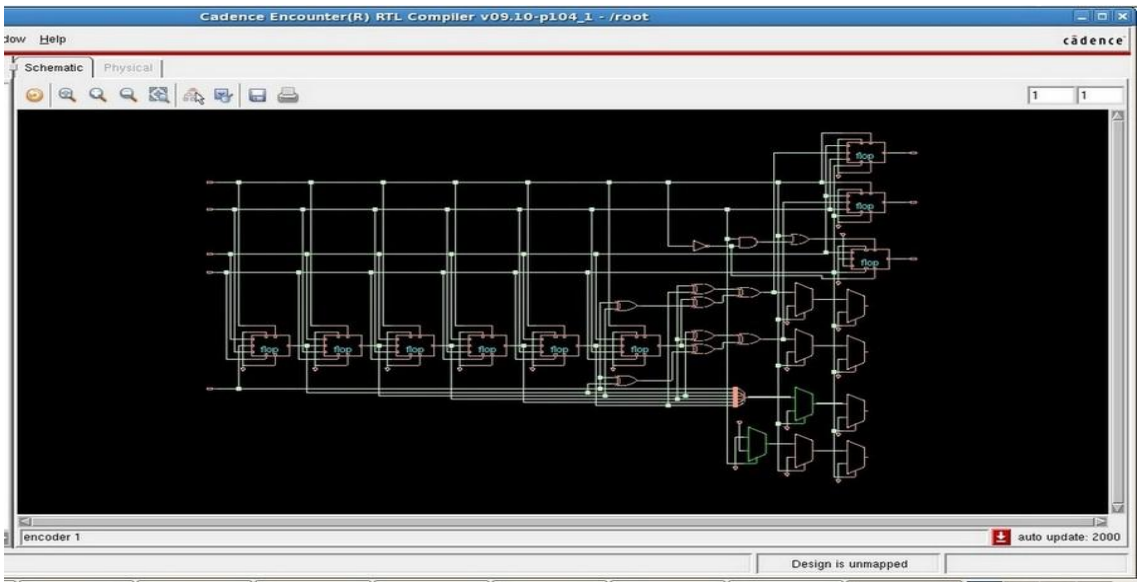

Fig 10: Synthesis Result of Convolutional Encoder

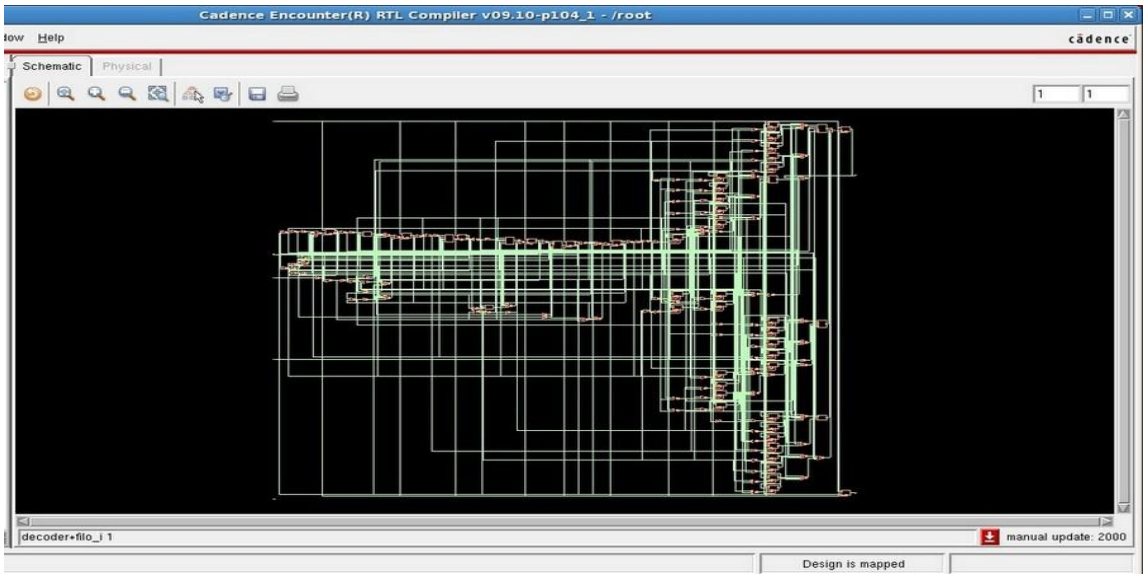

Fig 11: Synthesis Result of Viterbi Decoder

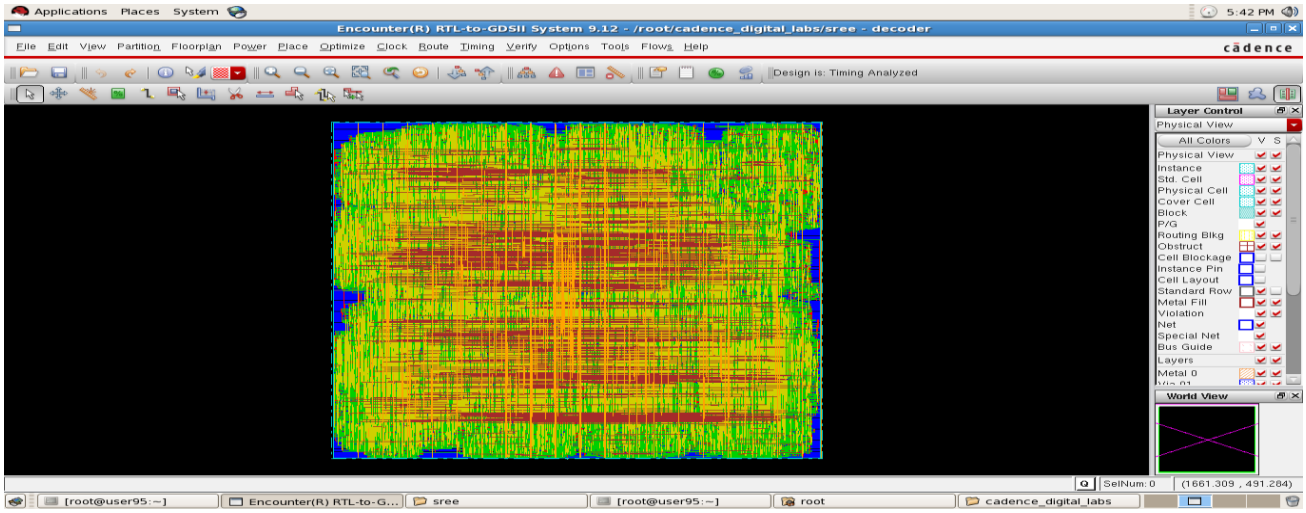

Fig 12: CMOS Standard Cell using 45nm Technology

Table II. Power Calculation of Viterbi Decoder

Instances	Number of Cells	Leakage Power (nW)	Internal Power (nW)	Net Power (nW)	Switching (nW)
Viterbi Decoder Using T-algorithm	63674	82118.33	69901503.63	9410461.65	79311965.29

6. CONCLUSION

We have proposed an efficient low power Viterbi decoder design using T-algorithm which reduces the power consumption and complexity of the Viterbi decoder without reducing the clock speed. The pre-computation steps are calculated. Both the architecture of the ACSU and SMU are modified. Which decodes the original signal the synthesis results of the convolutional encoder and decoder will be shown using Cadence RTL encounter tool. ASIC synthesis and power estimation results will be shown using TSMC 45nm CMOS Process. We proposed the low power scheme T-algorithm which reduces the power consumption and complexity without degrading the decoding speed.

7. REFERENCES

[1] "High Speed Low-Power Viterbi Decoder Design for TCM Decoder", IEEE Transaction on VLSI, VOL.20, NO.4, APRIL 2012.

[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, "T-algorithm decoding of channel convolutional 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.

[6] F. Chan and D. Haccoun, "Adaptive viterbi decoding of convolutional codes over memory-less channels," IEEE Trans. Commun., vol. 45, no. 11, pp. 1389–1400, Nov. 1997.

[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 decoder 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. 808–817, May-2010.