

# A Design of Efficient Viterbi Decoder with High Speed Low Power Consumption for TCM Decoders

Vidiyala Mallika<sup>1</sup>, B. Gajan<sup>2</sup>

VLSI Student, RMCEAC, Hyderabad, India<sup>1</sup>

HOD (ECE), RMCEAC, Hyderabad, India<sup>2</sup>

**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. We propose a pre-computation architecture incorporated with T-algorithm for VD, which can effectively reduce the power consumption 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-3/4 convolutional code used in a TCM system shows that compared with the full trellis VD, the pre computation architecture reduces the power consumption by as much as 70% without performance loss, while the degradation in clock speed is negligible.

**Keywords:** Trellis coded modulation (TCM), viterbi decoder, VLSI.

## I. INTRODUCTION

A Viterbi decoder uses the Viterbi algorithm for decoding a bit stream that has been encoded using convolutional code or trellis code. There are other algorithms for decoding a convolutionally encoded stream (for example, the Fano algorithm). The Viterbi algorithm is the most resource-consuming, but it does the maximum likelihood decoding. It is most often used for decoding convolutional codes with constraint lengths  $k \leq 10$ , but values up to  $k=15$  are used in practice. Viterbi decoding was developed by Andrew J. Viterbi and published in the paper "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", IEEE Transactions on Information Theory, Volume IT-13, pages 260-269, in April, 1967. There are both hardware (in modems) and software implementations of a Viterbi decoder.

Trellis coded modulation (TCM) schemes are used in many bandwidth-efficient systems. Typically, a TCM system employs a high-rate convolutional code, which leads to a high complexity of the Viterbi decoder (VD) for the TCM decoder, even if the constraint length of the convolutional code is moderate. For example, the rate-3/4 convolutional code used in a 4-D TCM system for deep space communications

[1] 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 convolutional code with a constraint length of 9 due to the large number of transitions in the trellis. Therefore, in terms of power consumption, the Viterbi decoder is the dominant module in a TCM decoder.

In order to reduce the computational complexity as well as the power consumption, low-power schemes should be exploited for the VD in a TCM decoder. General solutions for low-power VD design 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) [2],  $\square$ -algorithm [3] and  $\square$ -algorithm [4], [5]) or by over-scaling the supply voltage [6]. Over-scaling of the supply voltage usually needs to take into consideration the whole system that includes the VD (whether the system allows such an over-scaling or not), which is not the main focus of our research. RSSD is in general not as efficient as the  $\square$ -algorithm [2] and  $\square$ -algorithm is more commonly used than  $\square$ -algorithm in practical applications, because the  $\square$ -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. T-algorithm has been shown to be very efficient in reducing the power consumption [7], [8].

However, searching for the optimal PM in the feedback loop still reduces the decoding speed. To overcome this drawback, two variations of the  $\square$ -algorithm have been proposed: the relaxed adaptive VD [7], 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) [8]. In our preliminary work [9], we have shown that 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 PM and the accurate one. On the other hand, the SST based scheme requires predecoding and re-encoding processes 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. Therefore, the computational overhead and decoding latency due to predecoding and re-encoding of the TCM signal become high. In our preliminary work [9], we proposed an add-compare-select unit (ACSU) architecture based on precomputation for VDs incorporating  $\gamma$ -algorithm, which efficiently improves the clock speed of a VD with  $\gamma$ -algorithm for a rate-3/4 code. In this work, we further analyze the precomputation algorithm. A systematic way to determine the optimal precomputation 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-1/4 convolutional code [1]. Finally ASIC implementation results of the VD are reported, which have not been obtained in our previous work in [9]. The remainder of this paper is organized as follows. Section II gives the background information of VDs. Section III presents the precomputation architecture with  $\gamma$ -algorithm and discusses the choice of precomputation 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

The basic units of viterbi decoder are branch metrics, Add compare select and Survivor management unit. Figure 1 shows the general structure of a Viterbi decoder. It consist of three blocks: the branch metric unit (BMU), which computes metrics, the add-compare-select unit (ACSU), which selects the survivor paths for each trellis state, also finds the minimum path metric of the survivor paths and the survivor management unit (SMU), that is responsible for selecting the output based on the minimum path metric.

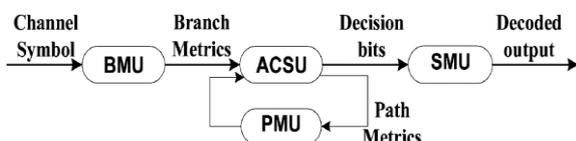


Fig. 1. Functional diagram of a viterbi decoder

Viterbi algorithm is called optimum algorithm since it minimizes the probability of error. The Viterbi algorithm can be explained briefly with the following three steps.

1. Weigh the trellis; that is, calculate the branch metrics.
2. Recursively compute the shortest paths to time n, in terms of the shortest paths to time n-1. In this step, decisions are used to recursively update the survivor path of the signal. This is known as add-compare-select (ACS) recursion.
3. Recursively find the shortest path leading to each trellis state using the decisions from Step 2. The shortest path is called the survivor path for that state and the process is referred to as survivor path decode. Finally, if all survivor paths are traced back in time, they merge into a unique path, which is the most likely signal path. The following example, describes the process of Conventional Decoding: Consider the received sequence as 11 10 00 01 01 11 which is error free and should be decoded. The step by step procedure for decoding convolutionally encoded data is given below. As soon as bits are received the following steps are repeated. Pair of Bits is given to the branch metric block which calculates the possible error metrics at that particular state. Two techniques used to find the possible error are hamming distance and Euclidean distance. The hamming distance is the number of bits not matching the possibility and the Euclidean distance is the point distance between the possibilities and the received data, which is obtained using the point distance formula. Hamming distance is selected as it is easy to implement on hardware.

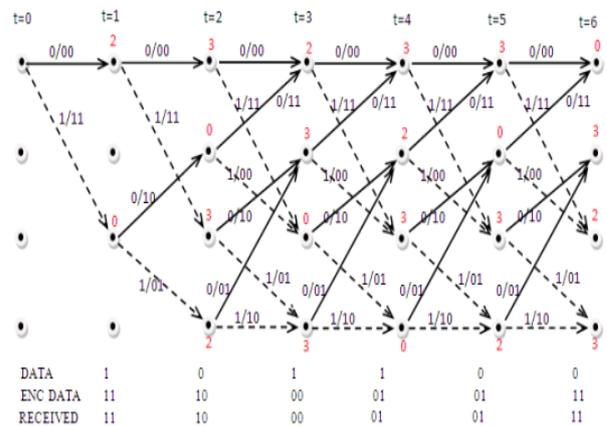


Fig 2 Trellis diagram for error free decoding

Any state from stage three in the trellis diagram can be reached from two possible previous states thus two error metrics is obtained. The Add compare select unit finds both the path metrics and compares, whichever is minimum that path metric is chosen as the new path metrics. If both path metrics are equal then any one is chosen and is stored in path metric matrix. Above two steps are repeated until the trellis ends and the entire path metric and next state metrics are obtained. Using these above matrices the survivor path traces the optimum path from last values of the next state matrix and then the data is decoded. The figure 2 shows the error free decoding of convolutional codes. Given the message data 101100 the encoded output is 11 10 00 01 01 11 which is received

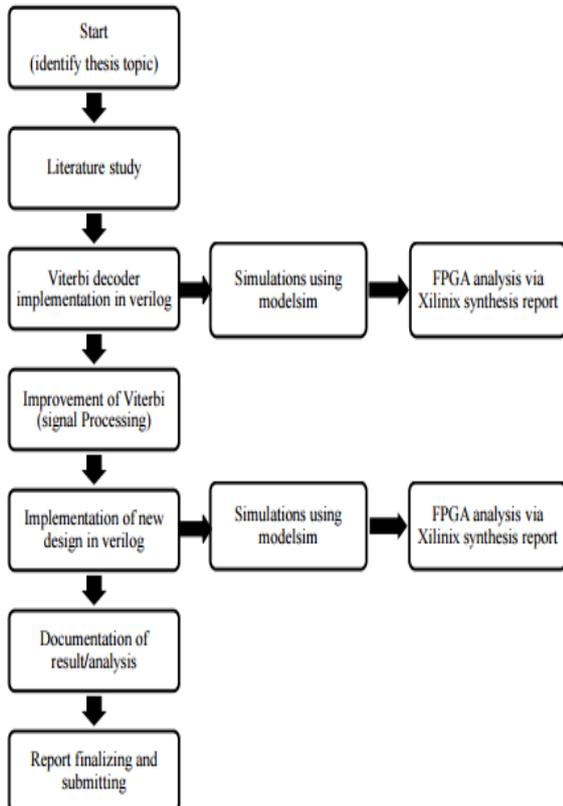
error free at the receiver. After completing the first two steps explained above the path metrics to reach each state in the trellis is obtained which is shown in red colour just above the states. After calculation of the path metrics.

There are two categories of coding methods:

- a. Backward error correction codes (BEC)
  - b. Forward error correction codes (FEC)
- Backward error correction codes are capable of the single task which is error correction. It is commonly used in the cases when the transmitted data is lost or corrupted, a request is made by the receiver to retransmit the data. BEC is also known as automatic repeat request (ARP). BEC is bandwidth efficient and mostly used in the systems with small data and minimum number of errors.
  - FEC is used for the improvement in the channel capacity. In this process we add some designed information to our data which is to be transmitted through the channel. For the improvement of channel different techniques are used one of them is channel coding. The performance of FEC is multi sender and multi path scenario depends heavily on the correlation of packet loss between paths

**III.FLOW CHART**

The working scheme of the thesis can be easily analyzed in the flow chart given below



**IV.CONVOLUTIONAL CODES PERFORMANCE WITH VITERBI ALGORITHM**

To implement a convolutional coding system, the appropriate code parameters, R and k, should be decided first. In addition, the design decisions for the decoding system, such as the quantization levels etc., are also crucial. This requires knowledge of the code performance with the particular parameters. The most useful techniques for estimating the performance of convolutional codes are union bounds and computer simulation

• Error event and union bounds

When decoding a convolutional code with Viterbi algorithm, an error event occurs at time i if the Viterbi decoder chooses an incorrect path which has a smaller Hamming distance over the correct path. This is shown in Figure 3

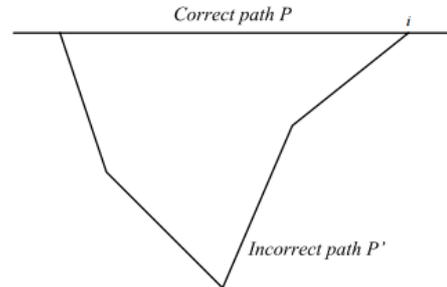


Fig. 3: The error event with Viterbi decoding

The probability of an error event,  $P_e$ , at time i is bounded by the sum of the error probabilities for all possible paths beginning and ending in the correct path.

$$P_e < \sum_{d=d_{free}}^{\infty} n_d P_d$$

where  $n_d$  is the number of possible paths of weight d merging with the all-zero path, and  $P_d$  is their probability. The  $d_{free}$  is the minimum Hamming distance between two paths which begin and end in the same state.

The union bound on the bit error rate,  $P_b$ , is obtained by summing the input sequence weights  $w_d$  corresponding to these paths, which is given by

$$P_b < \frac{1}{m} \sum_{d=d_{free}}^{\infty} w_d P_d$$

for a  $R = m/n$  code [7]. In practice, a finite number of  $w_d$  are computed to give an approximated union bound value. The accuracy of this approximation depends on the number of  $w_d$  samples used. The more the samples the closer the approximation is to the real value. Figure 2.14 compares the BER results from union bound calculation and simulation for  $R = 1/2$ ,  $k = 7$  code with hard-decision

Viterbi decoding. The line marked with crosses shows the BER from the union bound approximation with seven  $w_d$  samples.

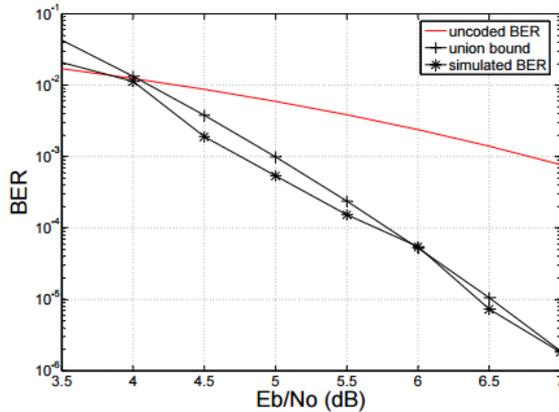


Fig. 4: Union bound and simulated BER for R=1/2, k=7 code with hard-decision Viterbi decoding and QPSK (Quadrature Phase Shift Keying) modulation.

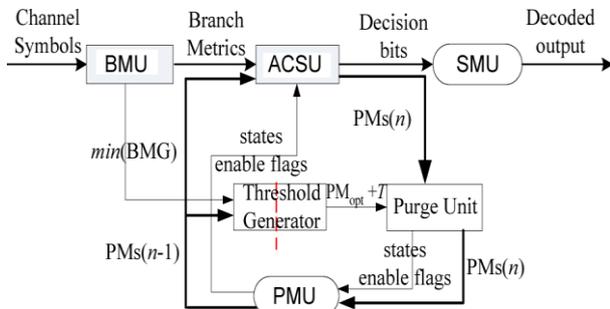


Fig. 5 VD with two-step precomputation T-algorithm

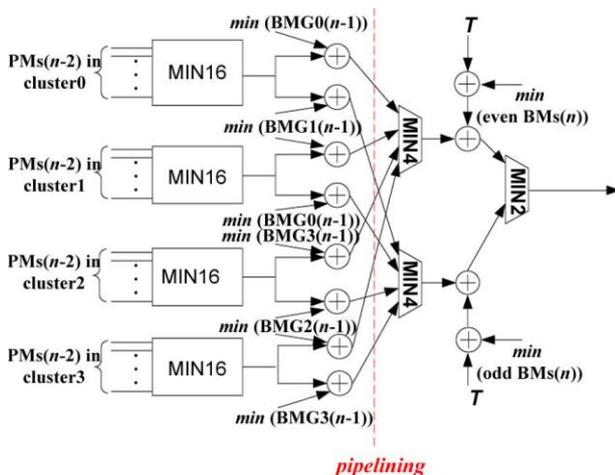


Fig. 6. Architecture of TGU

A. ACSU Design

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

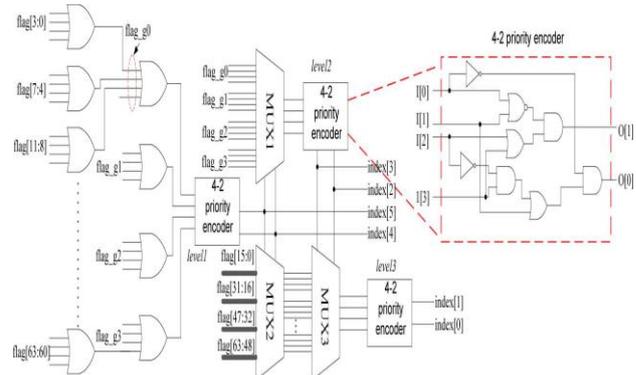


Fig. 7 Architecture of 64-to-6 priority encoder

The functional block diagram of the VD with two-step precomputation T -algorithm is shown in Fig. 5. 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 and the new PMs are then compared in the “Purge Unit” (PU). The architecture of the TGU is shown in Fig. 6, which implements the key functions of the two-step precomputation scheme. In Fig. 6, the “MIN 16” unit for finding the minimum value in each cluster is constructed with two stages of four-input comparators. This architecture has been optimized to meet the iteration bound [9]. Compared with the conventional T-algorithm, the computational overhead of this architecture is 12 addition operations and a comparison. B. SMU Design. In this section, we address an important issue regarding SMU design when -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 outputs 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 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 , which is always enabled, to output or trace back data. The process of searching for can find out the index of the optimal state as a byproduct. However, when the estimated is used [8], or in our case is calculated from PMs at the previous time slot, it is difficult to find the index of the optimal state.

TABLE I TRUTH TABLE OF 4-TO-2 PRIORITY ENCODER

input ( $I [3:0]$ )	output ( $O [1:0]$ )
x x x 1	0 0
x x 1 0	0 1
x 1 0 0	1 0
1 0 0 0	1 1

**TABLE II SYNTHESIS RESULTS FOR MAXIMUM  
CLOCK SPEED**

	Max speed (MHz)	cell area (mm <sup>2</sup> )
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 III POWER ESTIMATION RESULTS**

	Power (mw)	
Full-trellis VD	21.473 (100%)	
VD with 2-step pre-computation architecture	$T_{pm} = 0.75$	20.069 (93.5%)
	$T_{pm} = 0.625$	17.186 (80.0%)
	$T_{pm} = 0.5$	11.754 (54.7%)
	$T_{pm} = 0.375$	6.6127 (30.8%)

### V. IMPLEMENTATION RESULTS

The full-trellis VD, the VD with the two-step precomputation architecture and one with the conventional  $\square$ -algorithm 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 results are shown in Table III. Table III shows that the VD with two-step precomputation architecture only decreases the clock speed by 11% compared with the full trellis VD. Meanwhile, the increase of the hardware area is about 17%. The decrease of clock speed is inevitable since the iteration bound for VD with T-algorithm is inherently longer than that of the full-trellis VD. Also, any kinds of low-power scheme would introduce extra hardware like the purge unit shown in Fig. 5 or the clock-gating module in the SMU. Therefore, 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 the proposed architecture, which is counterintuitive. This is because the former decoder has a 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 applications. If the target throughput is moderately high, the proposed architecture can operate at a lower supply voltage, which will lead to quadratic power reduction compared to the conventional scheme. Thus we next focus on the power comparison between the full trellis VD and the proposed scheme. We

estimate the power consumption of these two designs with Synopsys Prime Power under the clock speed of 200 Mb/s (power supply of 1.0 V, temperature of 300 K). A total of 1133 received symbols (12 000 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. Therefore, 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 precomputation architecture that incorporates T-algorithm efficiently reduces the power consumption of VDs without reducing the decoding speed appreciably. We have also analyzed the precomputation algorithm, where the optimal precomputation steps are calculated and discussed. This algorithm is suitable for TCM systems which always employ high-rate convolutional codes. Finally, we presented a design case. Both the ACSU and SMU are modified to correctly decode the signal. ASIC synthesis and power estimation results show that, compared with the full-trellis VD without a low-power scheme, the precomputation VD could reduce the power consumption by 70% with only 11% reduction of the maximum decoding speed.

### ACKNOWLEDGMENT

It is my privilege and pleasure to express my profound sense of respect, gratitude and indebtedness to my guide **Mr. B. Gajan**, Head of the Department of ECE, Raja Mahendra College of Engineering, for his indefatigable inspiration, guidance, cogent discussion, constructive criticisms and encouragement throughout this dissertation work.

I extend my sincere thanks to **Mr. Naveen Kumar Mucha**, project adviser, Asst. Prof. in Electrical Dept for his encouragement and constant help.

### REFERENCES

- [1] "Bandwidth-efficient modulations," Consultative Committee For Space and System, Matera, Italy, CCSDS 401(3.3.6) Green Book, Issue 1, Apr. 2003.
- [2] 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.
- [3] C. F. Lin and J. B. Anderson, " $\square$ -algorithm decoding of channel convolutional codes," presented at the Princeton Conf. Info. Sci. Syst., Princeton, NJ, Mar. 1986.
- [4] S. J. Simmons, "Breadth-first trellis decoding with adaptive effort," IEEE Trans. Commun., vol. 38, no. 1, pp. 3–12, Jan. 1990.
- [5] F. Chan and D. Haccoun, "Adaptive viterbi decoding of convolutional codes over memoryless channels," IEEE Trans. Commun., vol. 45, no. 11, pp. 1389–1400, Nov. 1997.

- [6] 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.
- [7] 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.
- [8] F. Sun and T. Zhang, "Lowpower state-parallel relaxed adaptive viterbi decoder design and implementation," in Proc. IEEE ISCAS, May 2006, pp. 4811–4814.
- [9] 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.
- [10] 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.

### **BIOGRAPHY**



**Vidiyala Mallika** is an student in M.Tech (VLSI), from Rajamahendra college Hyderabad (JNTU-H) and Recieved her B.Tech degree from Pujya Srhri Madhavanji college hyderabad (JNTU-H) in ECE Dept.