iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** III **Month of publication:** March 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

WiMAX Channel Decoders with VLSI Architectures

Challa Vidya Tejaswini[1], P Kiran Kumar (B.Tech) [2]
[1]PG Student: Department of ECE, Anna University Regional Centre, Coimbatore, India
[2]ASIC Verification Engineer, Graphene Semiconductor Pvt., Ltd., Bangalore, India

Abstract - Polar codes have raised as an important error correction codes due to their capacity approaching property and also seen them as a major breakthrough in coding theory. WiMAX, which is a wireless communications standard, designed to provide high data rates. WiMAX has been adopted several channel codes at the decoding stage. Instead of those, it was suggested that polar codes which are designed for high speed, can be used. Therefore, a polar decoder which is efficient in terms of speed and with reduced hardware complexity is suggested. This proposed work is compared with different decoders, theoretically, exists for WiMAX. This shows that polar codes are highly superior to other codes in terms of high throughput and hardware complexity.
Index items- BTC decoder, CTC decoder, LDPC decoder, Polar codes, viterbi decoder, WiMAX,

I. INTRODUCTION

WiMAX (Worldwide Interoperability for Microwave Access) is a wireless communications standard which is designed to provide high data rates. The name "WiMAX" was created by the WiMAX Forum in order to promote interoperability and conformity of the standard. It includes the following potential applications: To provide portable mobile broadband connectivity across cities and countries through a variety of devices, to provide a wireless alternative instead of cable and digital subscriber line (DSL) for "last mile" broadband access to provide data, telecommunications, to provide a source of Internet connectivity as part of a business continuity plan. There are some comparisons and confusion between WiMAX and Wi-Fi are frequent, because both are related to wireless connectivity and Internet access. WiMAX is a long range system that covers many kilometers, which includes licensed or unlicensed spectrum to deliver connection to a network, simply the internet. Wi-Fi follows unlicensed spectrum to provide access to the local network and it is more popular in end user devices. WiMAX runs a connection-oriented MAC, whereas Wi-Fi runs on the Media Access Control's CSMA/CA protocol, which is connectionless and contention based. Both IEEE 802.16, which includes WiMAX, and IEEE 802.11, which includes Wi-Fi, define Peer-to-Peer (P2P) and wireless ad hoc networks, where an end user communicates to users or servers on another Local Area Network (LAN) using its access point or base station. On diffusing broadband wireless access systems, WIMAX has gained a wide popularity. In order to be reliable and flexible, WIMAX adopts several different channel codes, namely convolutional-codes (CC), convolutional-turbocodes (CTC), block-turbo-codes (BTC), low-density-parity-check (LDPC) codes, and recently introduced, polar codes (PC) that are able to cope with different application needs and channel conditions. On the other hand, high performance digital CMOS technologies have reached such a development that very complex algorithms can be implemented in low cost chips. The design of VLSI architectures for WIMAX channel decoders will be analyzed with emphasis on three main aspects: performance, complexity and flexibility. It was suggested to use polar decoders to achieve them.
Polar codes are linear block error correcting codes developed by “Erdal Arikan”. Different from other well-known capacity-approaching codes, such as Turbo codes and LDPC codes, polar codes are the first family of codes known to achieve channel capacity. Besides achieving the capacity for binary-input symmetric memory less channels, polar codes have also been proved to be able to achieve the capacity for any discrete and continuous memory less channel. They form a family of error correcting codes with an explicit and efficient construction encoding and decoding algorithms [4]. The organization of this paper is as follows. Section II discusses about the architectures of conventional decoders for WiMAX. Section III discusses about proposed work. Section IV concludes the paper by describing observations and also gives the scope of future work.

II. CONVENTIONAL DECODERS FOR WiMAX

A. Viterbi Decoders

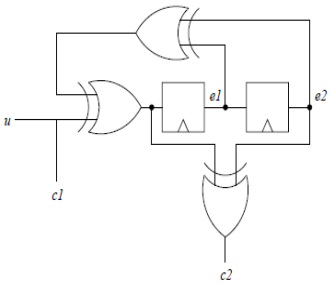
The most widely used algorithm to decode CCs is the Viterbi algorithm [10], which is based on finding the shortest path along a graph that represents the CC trellis. As shown in an example in figure. 1, a binary 4-states CC is shown as a feedback shift register (a) together with the corresponding state diagram (b) and trellis (c) representations. In the given example, the feedback

©IJRASET 2015: All Rights are Reserved

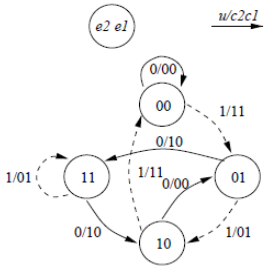
Volume 3 Issue III, March 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

shift register implementation of the encoder generates two output bits, c_1 and c_2 for each received information bit, u; c_1 is the systematic bit. The state diagram basically is a Mealy finite state machine describing the encoder behavior in a time independent way: each node corresponds to a valid encoder state, represented by means of the flip flop content, e_1 and e_2, while edges are labeled with input and output bits. The trellis representation also provides time information, explicitly showing the evolution from one state to another in different time. At each trellis step n, the Viterbi algorithm associates to each trellis state *S* a state metric Γ^S_n that is calculated along the shortest path and stores a decision d^S_n, which identifies the entering transition on the shortest path. First, the decoder computes the branch metrics (γ_n) [7], that are the distances from the metrics labelling each edge on the trellis and the actual received soft symbols. In the case of a binary CC with rate 0.5 the soft symbols are $\lambda 1_n$ and $\lambda 2_n$ and the branch metrics γ_n(c2,c1) (figure 2 (a)). Starting from these values, the state metrics are updated by selecting the larger metric among the metrics related to each incoming edge of a trellis state and storing the corresponding decision dSn.

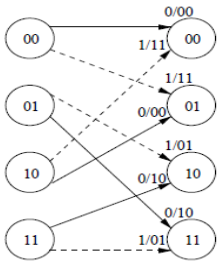
Finally, decoded bits are obtained by means of a recursive procedure usually referred to as trace-back. In order to estimate the sequence of bits that were encoded for transmission, a state is first selected at the end of the trellis portion to be decoded, then the decoder iteratively goes backward through the state history memory where decisions dSn have been previously stored: this allows one to select, for current state, a new state, which is listed in the state history trace as being the predecessor to that state.



(a)



(b)



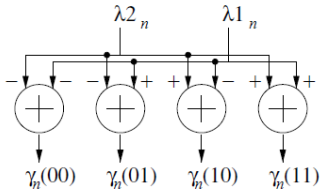
(c)

Figure: 1. Binary 4-state CC example: (a) shift register, (b) state diagram and (c) trellis representations

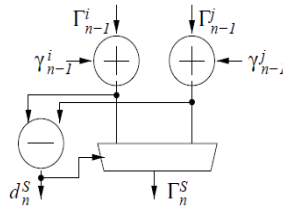
Different implementation methods are available to make the initial state choice and to size the portion of trellis where the trace back operation is performed: these methods affect both decoder complexity and error correcting capability. Looking at the global architecture, the main blocks required in aViterbi decoder is the branch metric unit (BMU) devoted to compute γ_n, the state metric unit (SMU) to calculate Γ^S_n and the trace-back unit (TBU) to obtain the decoded sequence. The BMU is made of adders and subtractors to properly combine the input soft symbols (see figure 2 (a)). The SMU is based on the so called add-compare select structure (ACS) as shown in figure.2 (b). This approach not only reduces the latency, but also the size of the decision memory that depending on the TBU radix.

©IJRASET 2015: All Rights are Reserved

Volume 3 Issue III, March 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)



(a)



(b)

Figure: 2. (a) BMU and (b) ACS architectures.

But it is necessary to improve the throughput [8], and it is suggested to use radix - 2, 4 as a choice for clock frequency of few tens to thousands of MHz.

B. BTC Decoders

Block Turbo Codes or product codes are serially concatenated block codes. Given two block codes C1=(n1,k1,δ1) and C2=(n2,k2,δ2) where ni, ki and δi represent the code-word length, the number of information bits, and the minimum Hamming distance, respectively, the corresponding product code is obtained according to [9] as an array with k1 rows and k2 columns containing the information bits. Then coding is performed on the k1 rows with C2 and on the n2 obtained columns with C1. The decoding of BTC codes can be performed iteratively row-wise and column-wise by using the sub-optimal algorithm. The basic idea relies on using the Chase search [3] a near-maximum-likelihood (near-ML) searching strategy to find a list of code-words and an ML decided code-word d={d0,…, dn-1} with dj ∈ {-1,+1}.

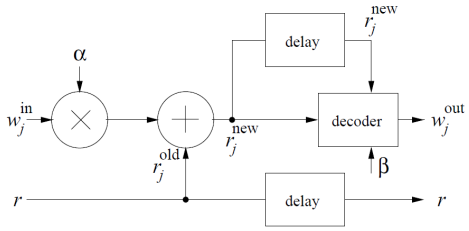


Figure: 3. Elementary block turbo decoder scheme.

A scheme of the elementary block turbo decoder is shown in figure. 3 where the block named “decoder” is a Soft-In-Soft-out (SISO) module that performs the Chase search. An effective solution to implement the SISO module is based on a three pipelined stage architecture where the three stages are identified as reception, processing, and transmission units. As detailed, during each stage, the N soft values of the received word r are processed sequentially in N clock periods. The reception stage is devoted to find the least reliable bits in the received code-word. The processing stage performs the Chase search and the transmission stage calculates λ(dj), wj and rj new. Another solution is proposed in [Goubier et al. 2008] where the elementary decoder is implemented as a pipeline resorting to the mini-maxi algorithm, namely by using mini-maxi arrays to store the best metrics of all decoded code-words in the Chase list. Due to its row-column structure, the block turbo decoder can be parallelized by instantiating several elementary decoders to concurrently process more rows or columns, thus increasing the throughput. The BTC specified for WIMAX is obtained using twice a binary extended Hamming code.

C. CTC Decoders

Convolutional turbo codes were proposed in 1993 by Berrou, Glavieux and Thitimajshima [2] as a coding scheme based on the parallel concatenation of two CCs by the means of an interleaver (Π) as shown in figure. 4 (a). The decoding algorithm is iterative and is based on the BCJR algorithm [1] applied on the trellis representation of each constituent CC (figure. 4 (b)). The key idea relies on the fact that the extrinsic information output by one CC is used as an updated version of the input a-priori

©IJRASET 2015: All Rights are Reserved

information by the other CC. As a consequence, each iteration is made of two half iterations, in one half iteration the data are processed according to the interleaver (Π) and in the other half iteration according to the deinterleaver (Π-1). The same result can be obtained by implementing in-order read/write half iteration and a scrambled (interleaved) read/write half iteration. The basic block in a turbo decoder is a SISO module that implements the BCJR algorithm in its logarithmic likelihood ratio (LLR) form.

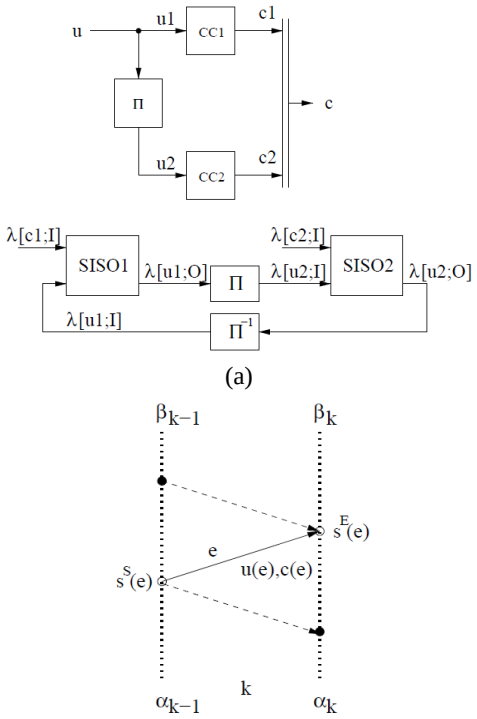


Figure: 4. Convolutional turbo code (a) coder and iterative SISO based decoder, (b) notation for a trellis step in the SISO.

The CTC specified in the WIMAX standard is based on a double binary 8-state constituent CC as shown in figure. 5, where each CC receives two uncoded bits (A, B) and produces four coded bits, two systematic bits (A,B) and two parity bits (Y,W). As a consequence, at each trellis step four transitions connect a starting state to four possible ending states.

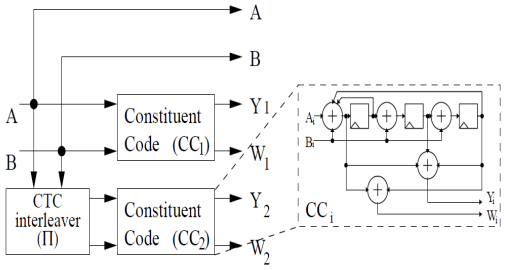


Figure. 5. WIMAX CTC: encoder and constituent CC structures.

Due to the trellis symmetry only 16 branch metrics out of the possible 32 branch metrics are required at each trellis step. As pointed out in high throughput can be achieved by exploiting the trellis parallelism, namely computing concurrently all the branch and state metrics. However, this technique requires estimating the circulation state at the decoder side. Since training operations to estimate the circulation state would increase the SISO latency, an effective alternative is to inherit these metrics from the previous iteration. As in Viterbi decoder architectures often in CTC decoders the state metrics are computed by means of the “wrapping” representation technique. It is worth pointing out that parallel architectures increase not only the throughput but also the complexity of the decoder, so that some recent works aim at reducing the amount of memory required implementing SISO local buffers.

©IJRASET 2015: All Rights are Reserved

D. LDPC Decoders

LDPC codes were originally introduced in 1962 by Gallager [5]. As turbo codes, they achieve near optimum error correction performance and are decoded by means of high complexity iterative algorithms. An LDPC code is a linear block code defined by a CxB parity check matrix H, characterized by a low density of ones: B is the number of bits in the code (block length), while C is the number of parity checks. A one in a given cell of the H matrix indicates that the bit corresponding to the cell column is used for the calculation of the parity check associated to the row. A popular description of an LDPC code is the bipartite (or Tanner) graph shown in Figure. 6 for a small example, where B variable nodes (VN) are connected to C check nodes (CN) through edges corresponding to the positions of the ones in *H*.

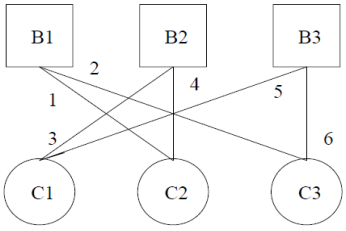


Figure: 6. Example Tanner graph

LDPC codes are usually decoded by means of an iterative algorithm variously known as sum-product, belief propagation or message passing, and reformulated in a version that processes logarithmic likelihood ratios instead of probabilities. In the first iteration, half variable nodes receive data from adjacent check nodes and from the channel and use those to obtain updated information sent to the check nodes; in the second half, check nodes take the updated information received from connected bit nodes and generate new messages to be sent back to variable nodes. In message passing decoders, messages are exchanged along the edges of the Tanner graph, and computations are performed at the nodes. To avoid multiplications and divisions, the decoder usually works in the logarithmic domain. Each variable node is initialized with the log-likelihood ratio (LLR) λ_j associated to the received bit. Next, messages are propagated from the variable nodes to the check nodes along the edges of the Tanner graph. At the first iteration, only λ_j are delivered, while starting from the second iteration VNs sum up all the messages Rij coming from CNs and combine them with λ_j. After a number of iterations that strongly depends on the addressed application and code rate (typically 5 to 40), variable nodes compute an overall estimation of the decoded bit.

A further change is usually applied to the scheduling of variable and check nodes in order to improve communications performance. In the two-phase scheduling, the updating of variable and check nodes is accomplished in two separate phases. On the contrary, the turbo decoding message passing (TDMP), also known as layered or shuffled decoding, allows for overlapped update operations: messages calculated by a subset of check nodes are immediately used to update variable nodes. This scheduling has been proved to be able to reduce the number of iterations by up to 50% at a fixed communications performance. The required number of functional units in a decoder can be estimated based on the concept of processing power Pc, which can be evaluated on the basis of the rate Rc of the code. Actually, most of the implementation concerns come from the communication structure that must be allocated to support message passing from bit to check nodes and vice versa. Several hardware realizations that have been proposed in the literature are focused on how efficiently passing messages between the two types of processing units. Three approaches can be followed in the high level organization of the decoder, coming to three kinds of architectures.

1) *Serial Architectures*: bit and check processors are allocated as single instances, each serving multiple nodes sequentially; messages are exchanged by means of a memory.
2) *Fully Parallel Architectures*: processing units are allocated for each single bit and check node and all messages are passed in parallel on dedicated routes.
3) *Partially Parallel Architectures*: more processing units work in parallel, serving all bit and check nodes within a number of cycles; suitable organization and hardware support is required to exchange messages.

For most codes and applications, the first approach results in slow implementations, while the second one has an excessive cost. As a result the only general viable solution is the third partially parallel approach, which on the other hand introduces the collision problem, already known in the implementation of parallel turbo decoders. Two main approaches have been proposed to deal with collisions: To design collision free codes, To design decoder architecture able to avoid or at least mitigate collision effects. However, structured LDPC codes, such as those specified in WIMAX, allow for replacing permutation networks by low complexity barrel shifters.

©IJRASET 2015: All Rights are Reserved

III. POLAR DECODER FOR WiMAX

A. Polar Decoders

In [4] Arikan has introduced the concept of polar codes and the main idea of polar coding is to create some coding system, so that one can easily access each coordinate channel in it and send data only through a particular channel for which the channel capacity is achieved to be maximum. The block lengths, N for these polar codes are restricted to powers of two, i.e. $N=2^n$, where $n \geq 0$. For a given polar code of length N, each code in the system is encoded in the same manner, like

$$x_1^N = u_1^N G_N \quad (1)$$

Let us consider a G_N code with parameters of N, K, A, u_{A^c} . Let the message, u_1^N be encoded into a codeword, x_1^N. Let the codeword x_1^N be sent over the channel, and let that channel output y_1^N be received. The main aim of the polar decoder's task is to generate an estimate $\hat{u}_1^N$ of u_1^N. Given any G_N code, polar decoder generates its decision $\hat{u}_1^N$ by computing,

$$\hat{u}_i \triangleq \begin{cases} u_i, & \text{if } i \in A^c \\ h_i(y_1^N, \hat{u}_1^{i-1}), & \text{if } i \in A \end{cases} \quad (2)$$

in the order i from 1 to N , and those decision functions are defined as

$$h_i(y_1^N, \hat{u}_1^{i-1}) \triangleq \begin{cases} 0, & \text{if } \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \geq 1 \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

It is obvious that decoder is said to have occurred with block error if the estimated outputs, $\hat{u}_1^N \neq u_1^N$.

The decision functions $\{h_i\}$ that are defined above resembles maximum likelihood decision functions but are not exactly so, because they treat the future frozen bits as random variables, rather than as known bits. In exchange for this optimality, decision function can be computed efficiently using recursive formulas. Apart from algorithmic efficiency, the recursive structure of the decision functions is important because it renders the performance analysis of the decoder tractable. Fortunately, the loss in performance due to not using true maximum likelihood decision functions happens to be negligible and the capacity is still achievable. The main block of polar decoder is shown in figure 7.

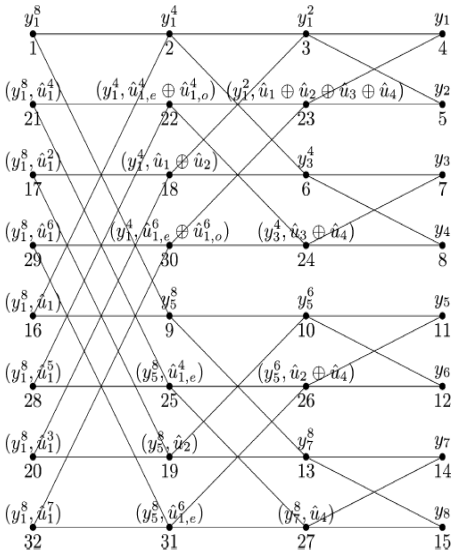


Figure: 7. Polar decoder for n=8 block length.

IV. CONCLUSION

In this paper, a survey has been done for WiMAX decoders. By comparing theoretically with the other decoders which uses CC, CTC, BTC, LDPC codes for WiMAX, the polar codes is said to have achieved channel capacity, high throughput and with reduced hardware complexity. Future work is to reduce latency and to improve error performance.

REFERENCES

[1] Bahl, L. R.; Cocke, J.; Jelinek, F. & Raviv, J, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate" , *IEEE Trans. on Information Theory*, Vol. 20, No. 2,, pp. 284-287, Mar. 1974.

©IJRASET 2015: All Rights are Reserved

[2] Berrou, C.; Glavieux, A. & Thitimajshima P, "Near Shannon Limit Error-Correcting Codes : Turbo Codes" , *Proceedings of the IEEE International Conference on Communications*, pp. 1064-1070, Geneva, Switzerland, May 1993.

[3] Chase, D, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information", *IEEE Trans. on Information Theory*, Vol. IT-19, No. 1, pp.170-182, Dec. 1972.

[4] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[5] Gallager R. G, " Low-Density Parity-Check Codes", *IRE Trans. on Information Theory*, Vol. 8, No. 1, pp. 21–28, Jan. 1962.

[6] Guilloud, F.; Boutillon, E.; Tousch, J. & Danger, J, "Generic description and synthesis of LDPC decoders", *IEEE Trans. on Communications*, Vol. 55, No. 11, pp. 2084-2091, Nov. 2007.

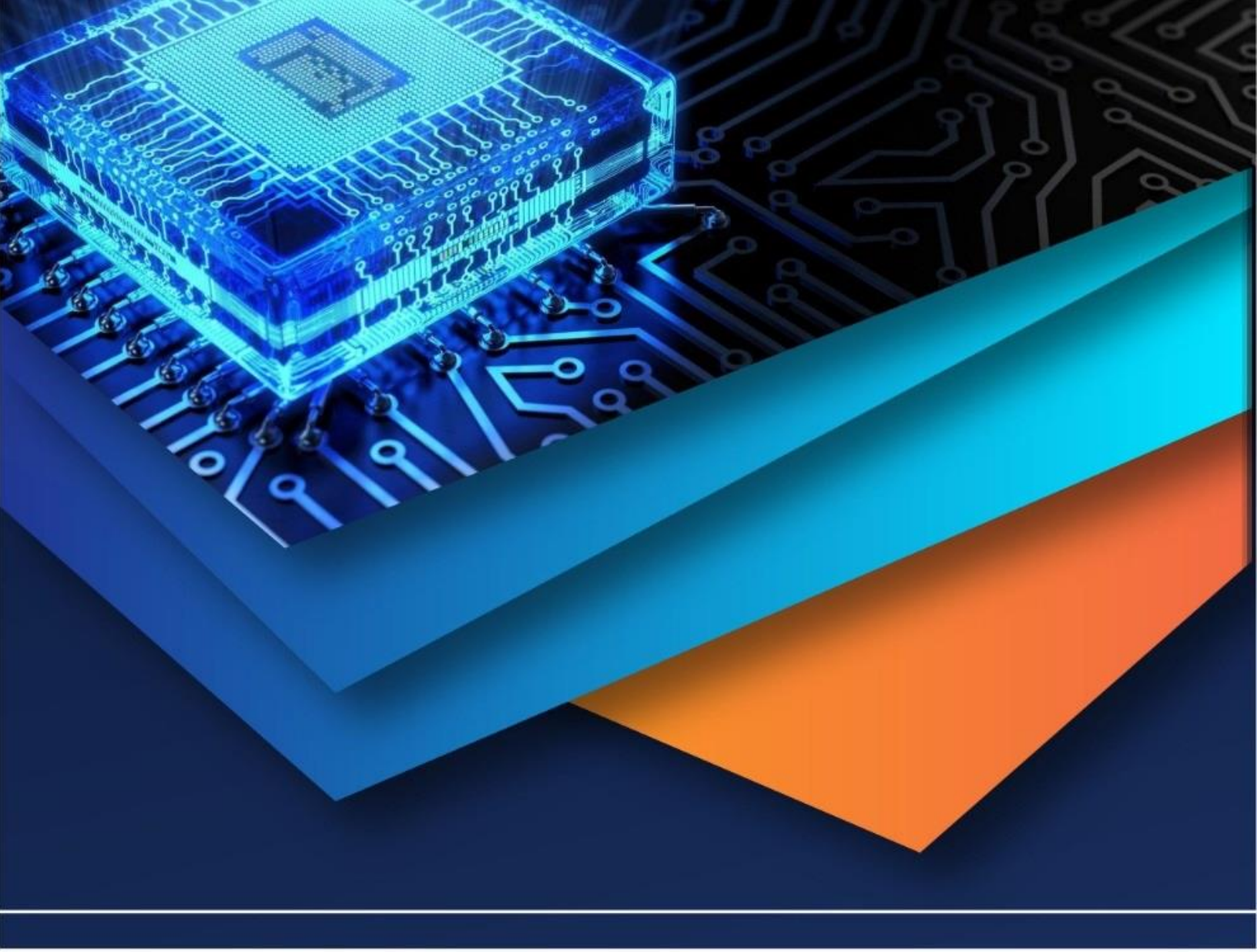
[7] Hekstra, A. P, "An Alternative to Metric escaling in Viterbi Decoders", *IEEE Trans. On Communications*, Vol. 37, No. 11, pp. 1220-1222, Nov. 1989.

[8] Kong, J. J. & Parhi, K. K., " Low-Latency Architectures for High-Throughput Rate Viterbi Decoders", *IEEE Trans. on VLSI*, Vol. 12, No. 6, pp. 642-651, Jun. 2004.

[9] Pyndiah, R. M., "Near-Optimum Decoding of Product Codes: Block Turbo Codes", *IEEE Trans. on Communications*, Vol. 46, No. 8, pp. 1003-1010, Aug. 1998.

[10] Viterbi, A. J., "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. on Information Theory*, Vol. IT 13, pp. 260-269, Apr, 1967.

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)