

Design of Low Power Dual Mode MLMAP Decoder Using Radix 4 Approach

S.Divya

PG Scholar, PSNA College of Engg. & Technology, Dindigul, India.

Dr.P.Maniraj Kumar

Professor, Dept.of ECE, PSNA College of Engg. & Technology, Dindigul, India.

Abstract – In order to have reliable communication, channel coding is often employed. Turbo code as a powerful coding technique has been widely studied and used in communication systems. The most significant achievement in coding theory are the turbo codes which are forward error correction codes. Due to the limitation of the battery life of wireless devices, transmitted power should remain as low as possible, which makes the system more susceptible to noise and interference. Error control coding is thus used to increase the noise immunity of the communication systems. The heart of the iterative decoding procedure is the use in each component decoder, of an algorithm that computes the a posteriori probability (APP) of the information symbols or more generally, a reliability value for each information symbol. Turbo decoder suffer from high decoding latency due to the iterative decoding process, the forward-backward recursion in the maximum a posteriori (MAP) decoding algorithm and the interleaving/de-interleaving between iterations. Computational complexity is estimated for the most popular Turbo decoding algorithms: MAP, Log-MAP, Max-Log-MAP and SOVA. Complexity studies showed that Max-Log-MAP is the best compromise between performance and complexity. Therefore, our implementation will be based on the Max-Log-MAP algorithm. ML MAP (Max Log Map) Architecture is proposed with SB & DB mode to achieve higher throughput and less power consumption. A Highly scalable LLR unit is proposed to increase the speed of the computation. The architecture is implemented using spartan3E family in Xilinx 12.1i.

Index Terms – Turbo decoder, interleaver, forward error correction codes, MLMAP decoder and LLR unit.

1. INTRODUCTION

With the rapid growth of multimedia services, forward error correcting (FEC) code has been a regular scheme for wireless communications to have a reliable transmission over noisy channels. Single-binary convolutional turbo code (SBCTC) proposed in 1993 has been the well-known FEC code that can achieve high coding gains close to the Shannon limit. The reduction of the CTC in bit error rate is achieved at the expense of intensive computations involved in the iterative turbo decoding steps and with the powerful algorithm called maximum a-posteriori probability (MAP) algorithm.

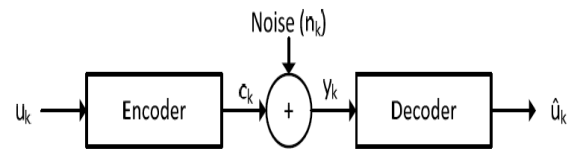


Figure 1 Model of digital communication system

Convolution coding is used in communication such as satellite and space communication to improve communication efficiency. The convolution code has enriched with the gift of a linear code with quality properties that it can operate on serial data as well as block codes. The convolution encoder with turbo decoder is a forward error correction method is mostly suited to a channel in which the transmitted signal is corrupted because of additive white Gaussian noise. IS-95, a wireless cellular standard for CDMA (Code Division Multiple Access), employs convolution coding which is the part of convolution coding. Convolutional encoding is a method of adding redundancy to a data stream in a controlled manner to give the destination the ability to correct bit errors without asking the source to retransmit. Convolutional codes, and other codes which can correct bit errors at the receiver, are called forward error correcting (FEC).

Turbo decoders are composed of two or more constituent SISO decoders, which correspond to the component codes employed in the transmitter, and an interconnection of these constituent decoders through an interleaver/deinterleaver. The decoding algorithm employed in the constituent decoders is the maximum a posteriori probability (MAP) algorithm. The MAP algorithm provides a reliability metric, known as the log-likelihood ratio (LLR), on the transmitted code symbols. The LLR output is employed by other constituent decoders, which attempt to improve their LLR estimates iteratively. However, the use of iterative processing results in a large computational and storage complexity and hence high power dissipation in the receiver. Therefore, low-power and high-throughput implementations for turbo decoders have recently been investigate for wireless and broadband applications.

2. TURBO DECODER

Turbo decoding is based on the principle of comparing the probability of a received soft input data being a '1' and '0'. The Turbo Decoder uses a decoding scheme called the MAP (Maximum A posteriori Probability) algorithm. The algorithm determines the probability of whether each received data symbol is a '1' as well as a '0'. This is done with the help of the data, parity symbols, and the decoder knowledge of the encoder trellis. A trellis is a form of a state transition table, of the encoder input/output]. Based on the data and parity information, the MAP decoder computes the probability of the encoder being in a particular state.

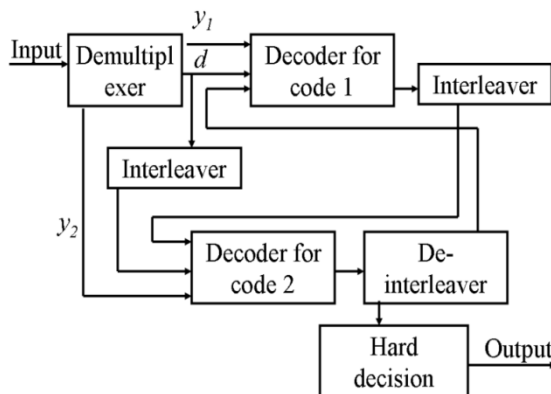


Figure 2 Iterative turbo decoder

Information is passed from one component decoder to another in the form of Log Likelihood ratio (LLR).

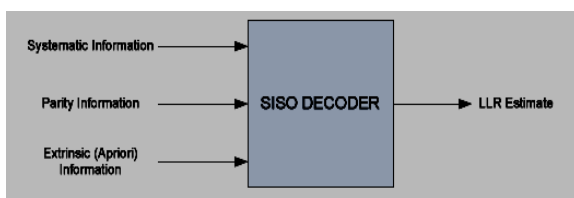


Figure 3 SISO decoder block diagram

As its name suggests, the LLR of a certain bit is the log of the ratio of the probability that a certain bit takes the value 1 to the probability that this same bit takes the value -1. The LLR of a bit d_k is denoted as $L(d_k)$ and is defined as,

$$L(d_k) = \log \left(\frac{P(d_k = +1)}{P(d_k = -1)} \right) \quad \dots \dots \dots (1)$$

Thus, taking the logarithm we will have a positive value if $P(d_k=1) > P(d_k=0)$, and negative value for the opposite. A positive value means the data value is a "1", otherwise a "0". For one complete cycle of iteration, one needs to compute the LLR using parity for non-interleaved as well as interleaved data.



Figure 4 main task of SISO decoder

3. TRELLIS DIAGRAM

A. Radix 2 Trellis Diagram

Trellis diagrams are somewhat complex than state and tree diagram but still they are most preferable for higher constraint length. The number of nodes at any level of the trellis does not continue to grow as the number of incoming message bits increase. It remains constant at 2^{k-1} state, where k is the constraint length of the code.

Figure 5 shows the trellis diagram of a rate 1/2 code with constraint length $K = 2$. The states are depicted as nodes, which denote the memory contents of the convolutional encoder. Each state has two branches, each of which corresponds to one of two possible input values. The branches of the trellis diagram are labeled with two bit branch words corresponding to the associated state transitions.

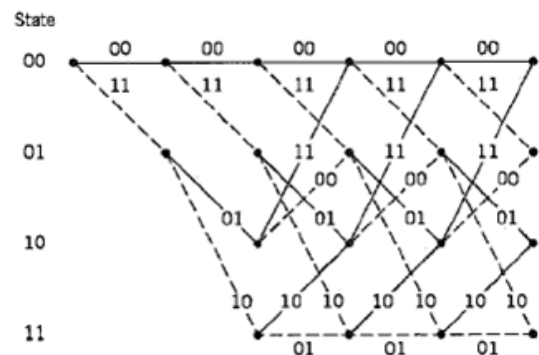


Figure 5 Code Rate 1/2 Radix 2 Trellis Diagram

B. Radix 4 Trellis diagram

The radix-4 butterfly is applied to the MAP decoder design to increase the decoding speed of the turbo decoder. Figure 4 shows the radix-4 trellis corresponding to the radix-2 trellis in Figure. 3. two stage radix-2 trellis can be merged to one stage radix-4 trellis by rearranging the states appropriately.

Notably, each branch of a radix-4 butterfly has 4-bit branch words, and corresponds to two input bits. Merging the trellis does not affect the selection of the most-likely bit, since a one-to-one mapping exists between the shortest path in the radix-4 trellis and the original radix-2 trellis. Each radix-4 butterfly in the radix-4 trellis has four origin and destination states, and 16 branches.

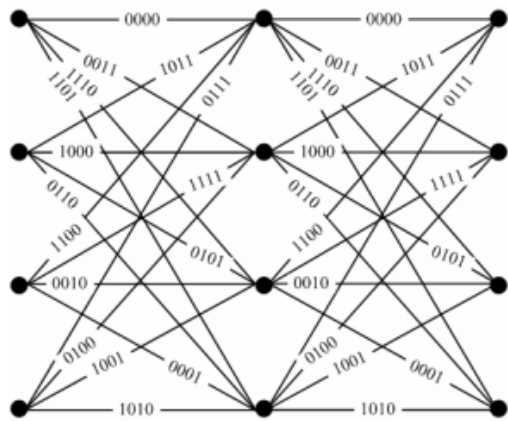


Figure 6 The radix-4 Trellis diagram

Figure 6 shows a radix-4 butterfly and its corresponding two stage radix-2 butterfly. Each branch of a radix-4 butterfly is a combination of two branches. For example, the branch word b_{00x00} in the radix-4 butterfly is combined from the branch words b_{00x0} at the first stage and b_{0x00} at the second stage, as shown in Figure. 7, and is given by $b_{00x00} = b_{00x0}b_{0x00}$. Similarly, the branch b_{10x01} in the radix-4 butterfly is combined from the branch b_{10x0} at the first stage and b_{0x01} at the second stage, and is given by $b_{10x01} = b_{10x0}b_{0x01}$.

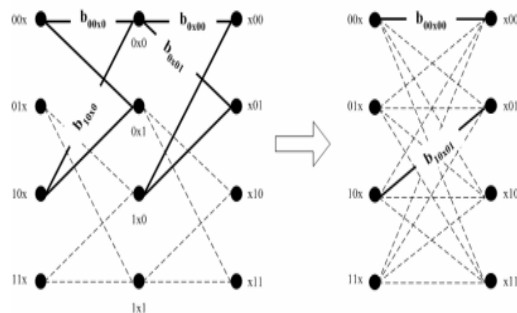


Figure 7 Radix 2 to radix 4 Trellis conversion

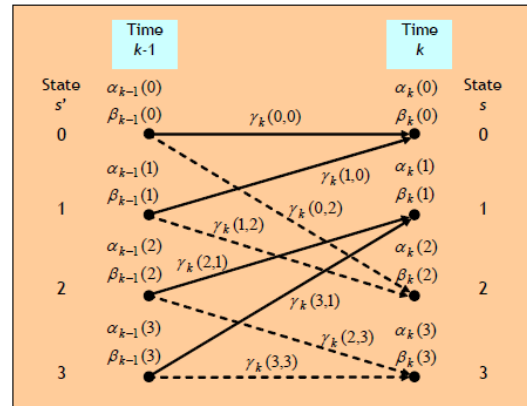
According to the symmetry, only the eight branch metrics need to be computed, and the other eight branch metrics, can be derived from the computed branches.

4. MAP DECODER

In order to implement an efficient turbo decoder, a suitable decoding algorithm has to be chosen. Turbo codes have been originally implemented with BCJR (Bahl, Cocke, Jelinek, Raviv) algorithm. However, this algorithm performs complex mathematical operations such as multiplication, division and logarithmic calculations. Therefore, engineers have avoided implementing this complex algorithm and preferred the sub-optimal derivatives of the BCJR (MAP) algorithm such as the Log-MAP and the Max-Log-MAP algorithms which are much simpler to implement but yield worse BER performances.

1. Representation Of A, B, Γ Values In Trellis Diagram

The trellis diagram parameters are represented by the following diagram in Figure.8.

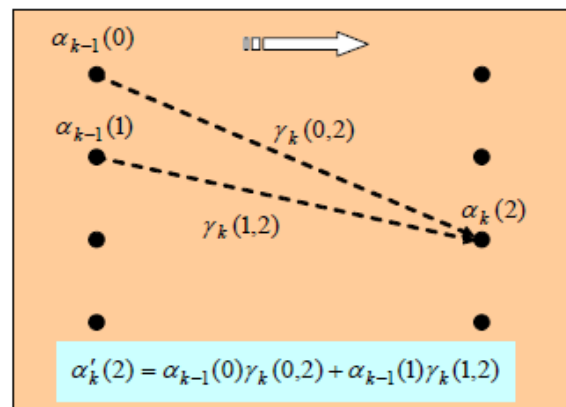
Figure 8 Labels of α , β , γ values

2. Calculation of Forward state metric values(α)

Let us begin with the recursive equation to obtain the “ α coefficients” of the BCJR algorithm according to eq. 2.

$$\alpha_k(m) = \frac{\sum_{m'} \sum_{i=-1}^{i=+1} \alpha_{k-1}(m') \gamma_i(R_k, m', m)}{\sum_m \sum_{m'} \sum_{i=-1}^{i=+1} \alpha_{k-1}(m') \gamma_i(R_k, m', m)} \dots \dots (2)$$

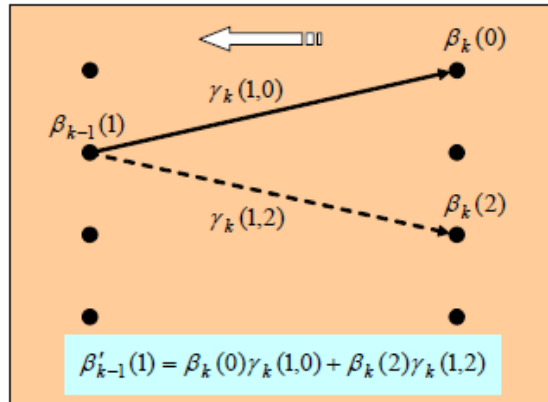
where $m = 0, 1, 2, \dots, M$, is the index of the states with $M = 2v - 1$.

Figure 9 Trellis Aided calculation of α values

3. Calculation of Backward state metric value(β)

As given in eq. 3, a similar procedure was followed to formulate the “ β coefficients” of the BCJR algorithm in matrix notation.

$$\beta_k(m) = \frac{\sum_{m'} \sum_{i=-1}^{i=+1} \beta_{k+1}(m') \gamma_1(R_{k+1}, m, m')}{\sum_m \sum_{m'} \sum_{i=-1}^{i=+1} \alpha_k(m) \gamma_1(R_{k+1}, m, m')} \dots \dots \dots (3)$$

Figure 10 Trellis Aided calculation of β values

4. Calculation of the Logarithmic Likelihood Ratios (LLRs)

Now, we will show how we can compute the logarithm of the likelihood ratios (LLR) associated with each bit d_k using the previously computed \mathbf{A} , \mathbf{B} and $\mathbf{\Gamma}$ matrices. LLR associated with each bit d_k is calculated as given in Eq.4.

$$\Lambda(d_k) = \ln \frac{\sum_m \sum_{m'} \gamma_{+1}(R_k, m', m) \alpha_{k-1}(m') \beta_k(m)}{\sum_m \sum_{m'} \gamma_{-1}(R_k, m', m) \alpha_{k-1}(m') \beta_k(m)} \dots \dots \dots (4)$$

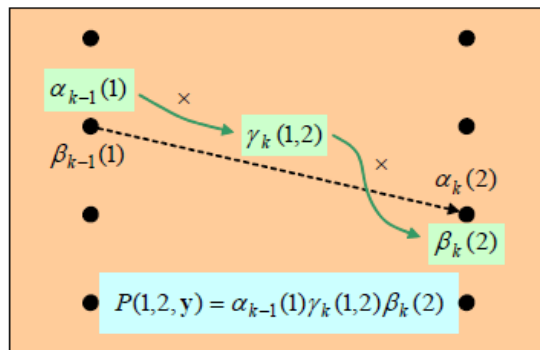


Figure 11 Basic step of LLR calculation

5. DUAL MODE SINGLE BINARY/DOUBLE BINARY (SB/DB) ML-MAP DECODING

The radix-4 SB and radix-4 DB MAP is reformulated to achieve high hardware usages and fully-shared storages of the dual mode MAP decoding. In this paper, the shared module design of dual-mode MAP is described in detail. Based on silicon area evaluations, the hardware overhead of dual mode MAP is less than 10% compared with the individual radix-4 SB MAP or individual radix-4 DB MAP.

B.RANCH METRICS DECOMPOSITION FOR RADIX-4 SB/DB MAP DECODING

For the both radix-4 SB and DB MAP decoding, the BMU and BMC designs become critical because 16 branch metrics are generated and stored. The decomposed branch metrics of the radix-2 SB MAP decoding was proposed to reduce the storages of four branch metrics. Partial data of the branch metrics generated by the BMU-S1 are stored into the BMC which is smaller than the conventional one. When the real branch metrics are required, they are calculated by the BMU-S2 with the stored partial data. In addition, the stored partial data are also used to generate the extrinsic information.

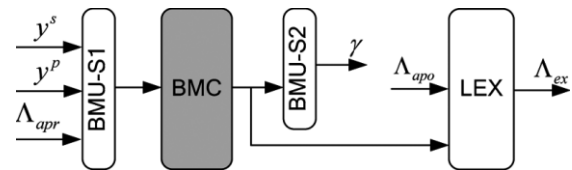


Figure 12 Block diagram of the branch metrics decomposition

B.RADIX-4 SB MAP DECODING

The SB CTC encodes one binary bit u_k at time k . For decoding two binary bits at a time, the radix-4 SB ML-MAP algorithm has been derived in by a look-ahead technique. The arithmetic operations of the radix-4 SB ML-MAP are described as follows.

$$\alpha_k(S_k) = \text{MAX}_{S_{k-1}, S_{k-2}} \left(\gamma_k(S_{k-2}, S_k) + \alpha_{k-2}(S_{k-2}) \right) \quad (5)$$

$$\beta_k(S_k) = \text{MAX}_{S_{k+1}, S_{k+2}} \left(\gamma_{k+1}(S_k, S_{k+2}) + \beta_{k+2}(S_{k+2}) \right) \dots \dots \dots (6)$$

$$\begin{aligned} \gamma_k(S_{k-2}, S_k) &= \Lambda_{apr, k-1}(u_{k-1}) + y_{s(k-1)} x_{s(k-1)} \\ &+ \sum_{i=1}^m y_{k-1}^{pi} x_{k-1}^{pi} \\ &+ (\Lambda_{apr, k}(u_k) + y_{sk}) x_{sk} \\ &+ \sum_{i=1}^m y_{pk} x_{pk} \dots \dots \dots (7) \end{aligned}$$

C.RADIX-4 DB MAP DECODING

In DB mode two binary bits are encoded $u_k = u_{1k}, u_{2k}$. The arithmetic operations of the of the Radix-4 DBMAP are described as

$$\alpha_k(S_k) = \text{MAX}_{S_{k-1}} \left(\gamma_k(S_{k-1}, S_k) + \alpha_{k-1}(S_{k-1}) \right) \quad (8)$$

$$\beta_k(S_k) = \text{MAX}_{S_{k+1}} \left(\gamma_{k+1}(S_k, S_{k+1}) + \beta_{k+1}(S_{k+1}) \right) \dots \dots \dots (9)$$

$$\gamma_k(S_{k-1}, S_k) = \Lambda_{\text{appr},k}^{(z)}(u_k = z) + 2\gamma_k^{s1} x_k^{s1} + 2\gamma_k^{s2} x_k^{s2} + 2 \sum_{i=1}^m \gamma_k^{pi} x_k^{pi} \dots \dots \dots (10)$$

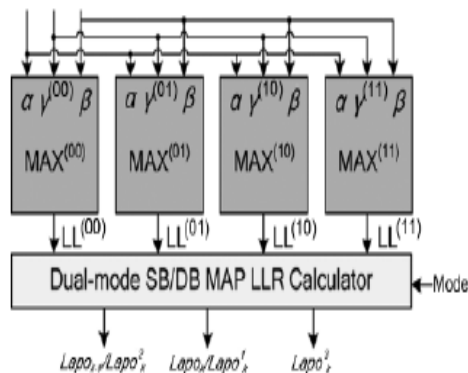


Figure 13 The block diagram of the LLR calculation in dual-mode ML MAP Decoder

6. RESULTS AND DISCUSSION

The various results of simulation are shown below. In Figure.14 the RTL schematic diagram of proposed dual mode SB/DB MLMAP decoder shows the no. of LUT's and other circuits used which is the conversion of high level language to schematic diagram.

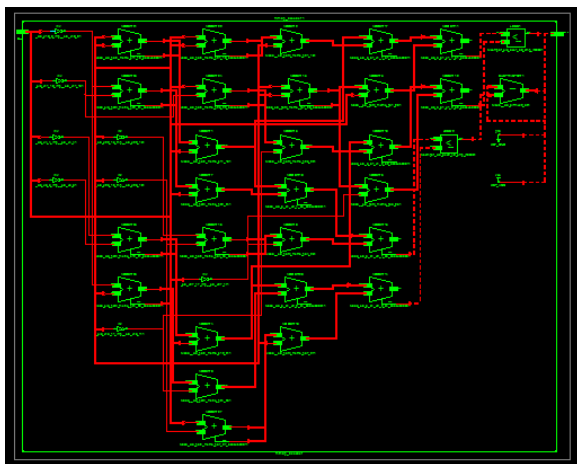


Figure 14 RTL schematic view

The Figure.15 shows the device utilization summary which says that the area utilization of the VLSI design. From which the percentage of area utilization can be viewed.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	43	29,504	1%
Number of occupied Slices	22	14,752	1%
Number of Slices containing only related logic	22	22	100%
Number of Slices containing unrelated logic	0	22	0%
Total Number of 4 input LUTs	43	29,504	1%
Number of bonded IOBs	15	250	6%
Average Fanout of Non-Clock Nets	3.70		

Figure 15 Device utilization summary of dual mode MLMAP decoder

Table 1 show the comparison results of the single binary and double binary mode area utilizations.

Table 1 Performance Analysis of Hardware Utilization

	SB MODE	DB MODE
Slices	110	89
LUT'S	70	60
Gate Counts	1789	1701
IOB's	21	21

7. CONCLUSION

In this paper, the performance of various decoding algorithm has been analyzed. Using the dual mode single binary/double binary MLMAP decoder, the high hardware sharing is achieved with reduced power consumption. The computation speed can be increased by using radix 4 trellis diagram. The computation of branch metric values can be reduced by branch decomposition method in radix 4 method. Due to the computation similarity present in both SB and DB method, the hardware sharing is achieved using multiplexer. The BER of this proposed method is better than the previous methods. The performance gap between the Max-Log-MAP and Log-MAP algorithms can be overcome by reduced computational complexity. The hardware shared dual mode MAP decoder achieves low computational cost and low memory storages.

REFERENCES

- [1] C. Shannon, "A Mathematical Theory of Information," Bell System Technical J., vol. 27, July 1948, pp. 379-423.
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes(1)," Proc. ICC-93, Geneva, Switzerland, May 1993, pp. 1064-1070.
- [3] M. M. Mansour and N. R. Shanbhag, "VLSI architectures for SISO-APP decoders," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 11, no. 4, pp. 627-650, Aug. 2003.
- [4] B. Bougard, A. Ciulietti, L. V. d. Perre, and F. Catthoor, "A class of power efficient VLSI architectures for high speed turbo-decoding," in Proc. IEEE Global Telecommunication Conf., vol. 1, 2002, pp. 553-549.
- [5] Z. Wang, Z. Chi, and K. K. Parhi, "Area-efficient high-speed decoding schemes for turbo decoders," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 10, no. 6, pp. 902-912, Aug. 2002.

- [6] Zohngfeng Wang and Xinming “ High Spped VLSI Architectutrs for Turbo Decoders “ in IEEE Transaction on very large integration (VLSI) System, Vol. 8, No. 1, February 2000.
- [7] E. Boutillon, W. J. Gross, and P. G. Gulak, “VLSI architectures for the MAP algorithm,” IEEE Trans. Commun., vol. 51, pp. 175–185, Feb. 2003.
- [8] B. Bougard, A. Ciulietti, L. V. d. Perre, and F. Catthoor, “A classof power efficient VLSI architectures for high speed turbo-decoding,” inProc. IEEE Global Telecommunication Conf., vol. 1, 2002, pp. 553–549.
- [9] Atar, O., Sazli, M.H., Ilk, H.G., “FPGA Implementation of Turbo Decoders”, KTT0 2011 11th International Conference on Knowledge in Telecommunication Technologies and Optics, pp. 103-108, Szczyrk, Poland, June 22-24 2011.
- [10] Wong, Y., Jian, W., HuiChong, O., Kyun, C., Noordi, N. “Implementation of Convolutional Encoder and Viterbi Decoder using VHDL”, Proceedings of IEEE International conference on Research and Development Malaysia, November 2009
- [11] S. Shiyamala and V. Rajamani, “A Novel Area Efficient Folded Modified Convolutional Interleaving Architecture for MAP Decoder,” International Journal of Computer Applications, Vol. 9, No. 9, pp-18-22, Nov. 2010.
- [12] S. M. Karim and IndrajitChakrabarti, “An Improved Low-Power High-Throughput Log-MAP Turbo Decoder”, IEEE Transactions on Consumer Electronics, Vol. 56, No. 2, May 2010.
- [13] J. Kaza and C. Chakrabarti, “Design and implementation of low-energy turbo decoders,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., Vol. 12, no. 9, pp. 968–977, Sep. 2004
- [14] M. Bickerstaff, L. Davis, C. Thomas, D. Garrett, and C. Nicol, “A 24Mb/s radix-4 LogMAP turbo decoder for 3GPP-HSDPA mobile wireless,” in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), 2003, pp. 1–10.