VLSI Implementation of Low Power High Speed ML MAP Processor Design for Dual Mode Binary Turbo Decoders

P.Maniraj Kumar[#1],Dr.S.Sutha[*2]
[#] Department of Electronics and Communication Engineering,
PSNA College of Engineering and Technology, Dindigul – 624 622 , India
[*]Department of Electrical and Electronics Engineering,
University College of Engineeirng, Panruti – India.
[1] mani1376@yahoo.com
[2] suthapadmanabhan@gmail.com

***Abstract*—This paper presents the Max Log Maximum a Posteriori (MAX Log MAP) architecture which influences maximum error correcting capacity and minimum hardware complexity. In Turbo decoding the design of Log MAP algorithm is complicated due to the exponential factor in the Log Likelihood Calculation. In this paper, a new architecture is designed based on jacobian logarithm which has a max function with the exponential correction factor. The complexity in the jacobian logarithm is further reduced by the Max Log Map algorithm by eliminating the correction term. Here Multistep Log Map algorithm is used for the approximation of the correction function. Furthermore a Dual Mode Single Binary and Double Binary (SB/DB) procedure is also used to compute the log likelihood ratio in order to reduce the hardware complexity and increase the speed of the computation. This method is implemented by XCS3S500E FPGA processor to achieve a High Speed 15% to 25 % (as in existing); with less area Utilization (8% as in existing) and low Power (74.95mW).The decoder throughput can be maintained in the proposed scheme without degrading the BER performance.**

Keyword: Max Log MAP, Log Likelihood Ratio, Single Binary and Double Binary and Jacobian Logarithm.

A. I INTRODUCTION

In 1948, Shannon introduced a theorem for channel codes. Based on this various applications have been developed [1]. Berrou proposed turbo codes in 1993, shows error correcting performance close to the Shannon limit [2]. This turbo code has been used in some applications where real-time processing is not required. For satellite communications, due to its hardware complexity and decoding delay, these turbo codes are not used. Since then, a research effort has been taken to improve the performance of the Turbo code. As a result, the Turbo code has been adopted in the IMT-2000 system for high data rate transmission.

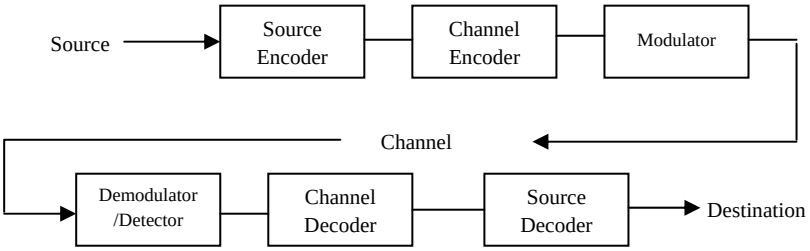


Fig1. Digital Communication System

Fig 1 shows the basic digital communication model. First three blocks of the diagram consisting of source encoder, channel encoder, and modulator forms the transmitter. A binary interface established between source and channel. The source encoder converts the source output to a binary sequence and the channel encoder processes the binary sequence for transmission over the channel. The last three blocks consisting of detector/demodulator, channel decoder, and source decoder form the receiver. The source decoder's responsibility is to retrieve the information from the channel decoder. Turbo code plays a vital role in such applications.

The basis of turbo coding is to introduce redundancy in the data to be transmitted through a channel [3]. The redundant data helps to recover original data from the received data. In data transmission, turbo coding helps to achieve near Shannon limit performance. Turbo-Codes operate on block level that is the data is separated into blocks. A block is decoded in iterative procedure by maximum a posteriori (MAP) or soft output Viterbi

algorithm (SOVA) component decoders. The turbo encoder transmits the encoded bits which form inputs to the turbo decoder. The turbo decoder decodes the information iteratively. The maximum a posteriori algorithm (MAP) is used in the turbo decoder.

1.1Turbo Decoders

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 Aposteriori 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 [4] [5]. Based on the data and parity information, the MAP decoder computes the probability of the encoder being in a particular state.

The decoding is accomplished by first demultiplexing the incoming data stream into D, M1, M2. Figure 2 represents a schematic diagram of turbo decoder. D and M1 go into the decoder as the first code1. This gives an estimate of the extrinsic information from the first decoder which is interleaved and pass on to the second decoder. The second decoder thus has three inputs, the extrinsic information from the first decoder, the interleaved data D, and the received values for M2. It produces its extrinsic information and this is deinterleaved and passed back to the first encoder. This process is then repeated or iterated as required until the final solution is obtained from the second decoder interleaver.

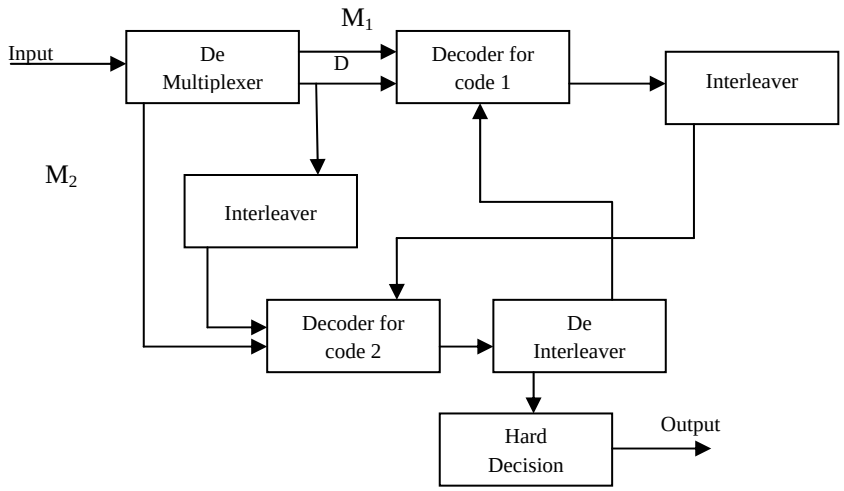


Fig 2. Schematic diagram of Turbo Decoder

The Log Likelihood Ratio (LLR) is the probability that the received data bit is a '0' divided by the probability that the received data bit is a '1'.

$$\text{L (D)} = \log \frac{P(D=0)}{P(D=1)}$$

Thus, taking the logarithm we will have a positive value if P (D=1) > P (D=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.

A Turbo encoder generates the parity bits (Ps) along with the input data (Xs) using two encoders. Both these encoders encode the input data and the modified data (Xs, Ps) is passed through an interleaver. A Turbo decoder consists of two decoders to perform iterative decoding by taking a-priori information from each decoder for a optimum error correction capability. The most widely used SISO decoding algorithm is maximum a posteriori probability (MAP) algorithm. The decoder implementing the MAP algorithm is more complex than other decoders, but the BER performance is twice as that of other decoders.

Owing to the recent advancement in fabrication and circuit design technology, the BER performance has become a key parameter than hardware complexity. As a result, the MAP algorithm is preferred in most decoders. A log-MAP decoding algorithm has been proposed for better error correction capability. A lot of methods are proposed for the calculation of correction values. But Log MAP algorithm is more complicated with the exponential sum. The complexity in the Log MAP algorithm is reduced by the Max Log Map algorithm by eliminating the correction term.

This paper proposes an algorithm to compute the correction function that will be used to calculate the state metrics and log likelihood values. Furthermore this Log Likelihood values can be modified into SB/DB mode with a new set of equations and this will increase the performance parameters of the Turbo decoders.

The rest of the paper is organized as follows: In section II, brief introduction of various MAP algorithms and different methods of approximation of the correction function are discussed. In Section III, the architecture of State Metrics and Branch Metric are proposed. In Section IV, the SB/DB mode architecture for LLR calculation is proposed. Section V presents the results and discussions. This paper is finally concluded in the Section VI.

B. II INTRODUCTION TO VARIOUS MAP ALGORITHMS

A. MAP Algorithm

MAP decoders are superior with respect to communications performance and for that reason preferred in advanced implementations. The MAP decoding algorithm is a recursive technique that computes the Log-Likelihood Ratio (LLR) of each bit based on the entire observed data block of length K. The MAP algorithm is used to calculate Log Likelihood Ratio (LLR) ($\Lambda(D_k)$) of the a posteriori probability (APP) of $D_k = 1$ to $D_k = 0$.

The state of the encoder is represented by S_k at time k. The state transition is from k-1 to k. The LLR value can be given by equation 1.

$$\Lambda(D_k) = \ln \frac{\sum_{S_k} \sum_{S_{k-1}} \gamma_1(y_k, S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})\beta_k S_k}{\sum_{S_k} \sum_{S_{k-1}} \gamma_0(y_k, S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})\beta_k S_k} \quad (1)$$

The forward state metric α_k can be expressed as

$$\alpha(S_k) = \frac{\sum_{S_{k-1}} \sum_{i=0}^{1} \gamma_i(y_k, S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})}{\sum_{S_{k-1}} \sum_{i=0}^{1} \gamma_i(y_k, S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})} \quad (2)$$

and the backward state metric β_k can be expressed as

$$\beta(S_k) = \frac{\sum_{S_{k+1}} \sum_{i=0}^{1} \gamma_i(y_{k+1}, S_k, S_{k+1})\beta_{k+1}(S_{k+1})}{\sum_{S_k} \sum_{S_{k+1}} \sum_{i=0}^{1} \gamma_i(y_{k+1}, S_k, S_{k+1})\alpha_k(S_k)} \quad (3)$$

The branch transition probabilities γ_i can be calculated by the equation (4).

$$\gamma_i((y_{sk}, y_{pk}), S_{k-1}, S_k) = e^{\sqrt{x_{sk}(L_e + (x_{sk}) + L_c(y_{sk}) + L_c y_{sk} x_{pk}}} \quad (4)$$

The channel reliability value $L_c = 2 / \sigma^2$ with σ^2 is the noise variance and L_e is the extrinsic information which gives a priori information.

B. *Log MAP Algorithm*

The implementation of MAP decoders in VLSI is more complex because of the recursive calculations. As pointed out in [10], it is mandatory to implement the MAP algorithm in the logarithmic domain (Log-MAP) to avoid numerical problems without degrading the decoding performance. So, this complexity in the MAP algorithm can be reduced by Log MAP algorithm. To avoid complicated calculations in the MAP algorithm, the entire MAP algorithm is calculated in Logarithmic domain. So the basic equations in the MAP algorithm (2), (3) and (4) are converted into log domain which reduces the number of strong operators such as multiplications and adds the weaker operations such as additions used.

So, $\alpha_k(S_k) = \ln \alpha_k(S_k)$.

$$\overset{\Lambda}{\alpha}(S_k) = \ln\Big(\sum_{S_{k-1}} \sum_{i=0}^{1} e^{\ln(\gamma_i(y_{sk}, y_{pk})S_{k-1}, S_k) + \ln \alpha_{k-1}(S_{k-1})}\Big) - \sum_{S_k} \sum_{S_{k-1}} e^{\ln(\gamma_i(y_{sk}, y_{pk})S_{k-1}, S_k) + \ln \alpha_{k-1}(S_{k-1})} \tag{5}$$

$$\overset{\Lambda}{\beta}(S_k) = \ln\Big(\sum_{S_{k+1}} \sum_{i=0}^{1} e^{\gamma_i(y_{S(k+1)}, y_{p(k+1)}, S_k, S_{k+1})\beta_{k+1}(S_{k+1})}\Big) - \sum_{S_k} \sum_{S_{k+1}} \sum_{i=0}^{1} e^{\ln(\gamma_i(y_{S(k+1)}, y_{p(k+1)}, S_k, S_{k+1})\alpha_k(S_k))} \tag{6}$$

$$\overset{\Lambda}{\alpha}(S_k) = \ln\Big(\sum_{S_{k-1}} \sum_{i=0}^{1} e^{\ln(\gamma_i(y_{sk}, y_{pk})S_{k-1}, S_k) + \ln \alpha_{k-1}(S_{k-1})}\Big) - \sum_{S_k} \sum_{S_{k-1}} e^{\ln(\gamma_i(y_{sk}, y_{pk})S_{k-1}, S_k) + \ln \alpha_{k-1}(S_{k-1})} \tag{7}$$

The above equation (5),(6) and (7) can be simplified into general form, which is given by the equation (8).

$$F(x_1, x_2, x_3 \ldots x_N) = \ln\Big(\sum_{i=1}^{n} e^{x_i}\Big) \tag{8}$$

The jacobian logarithm for two variable expressions is given by

$$\max^*(x_1, x_2) = \ln(e^{x1} + e^{x2}) = \max(x_1, x_2) + \ln(1 + e^{-|x_1 - x_2|}) = \max(x_1, x_2) + C(x) \tag{9}$$

The calculation of the correction term C(x) in (9) is more complicated in Log MAP algorithm. The same form of log exponential sum occurs in the calculation of backward state metrics (β_k) , and LLR Λ (D_k).So, to improve the performance of the Log Map Algorithm, a simple method of implementation of correction function must be required. Various methods of correction implantation is discussed in [6][7][8][9]. The easiest method of finding out the correction function is presented in [10].But this requires more memory to store the table sequences.

C. Max Log MAP Algorithm

The max log map algorithm introduces an approximation $\max^*(x_1,x_2) \approx \max(x_1,x_2)$. This Max log map algorithm simply omitting the correction term C(x).But the performance of the Log Map algorithm is dropped by 10% compared to Log Map algorithm. Even though the Max Log Map algorithm is least complex, it gives worst BER. So, in Max Log Map algorithm, a simple implementation of correction function is required to improve the performance of the turbo decoders.

D. Different Algorithms for Correction Approximation

In the Linear Log Map algorithm proposed in [8], the MacLaurin Series expansion is used and it is observed that the correction function is approximately zero.

$$C(x) \approx \max\left(0, \ln 2 - \frac{1}{2}x\right) \tag{10}$$

This approximation offers a good result than the Constant Log Map Algorithm presented in [9] where C(x) =

$$\begin{cases} \frac{3}{8}, -2 \le x \le 2 \\ 0, otherwise \end{cases} \tag{11}$$

An accurate method of approximation proposed in [10] is Multistep Log Map Algorithm.

$$C(x) \approx \frac{\ln 2}{2^{[x+0.5]}} \tag{12}$$

This method of correction is more accurate. This method of algorithm employs with shift registers storing the ln(2).For fast computation , a speed registers are required for this algorithm. Figure 3 shows the graphical comparison of various Approximations to the correction function.

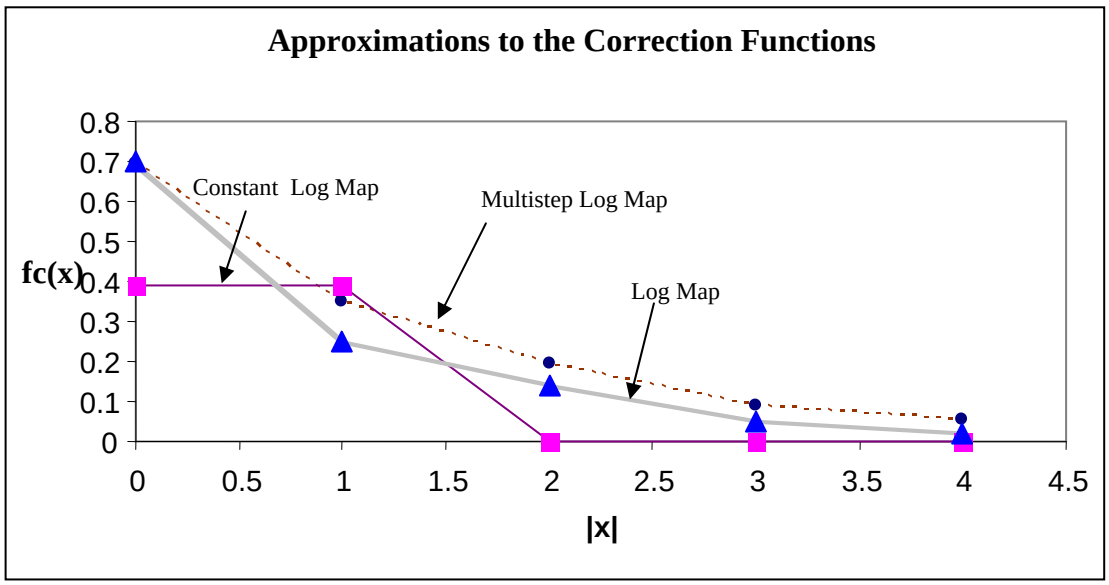


Fig. 3. Comparison of Various Correction functions.

III. PROPOSED LLR UNIT

A. *Piplined Metric Architecture*

In order to compute the LLR values forward (α) and backward (β) states, and branch metric (γ) values of all states are required. Figure 4 shows the metric pipelined architecture unit. This proposed architecture consists of three adders, a subtractor, a comparator, one Selection MUX, Trellis MUX and logic. Since, the add-compare select operation is more complex than Trellis MUX, a pipelined architecture proposed as shown in Fig. 4.

B. *Dual Mode Single Binary/Double Binary (SB/DB) MAP Decoding*

The SB/DB Map decoding is introduced by Y.Sun [11] and C.Y Shen [12] for radix 4 Map architecture to achieve high hardware usage and shared storage information. In general, the MAP is composed of branch metrics, forward recursion state metrics, backward recursion state metrics, a priori LLR, a posteriori LLR, and extrinsic information. The shared hardware architecture is implemented in SB and DB mode[16][17]. The equations of branch metrics, forward state metrics, backward state metrics; a priori LLR, a posteriori LLR, and extrinsic information are reformulated for SB / DB Decoding. [18][19][20].

i) Radix-4 SB MAP Decoding

The arithmetic operations of the of the Radix-4 SB MAP are described as

$$\alpha_k(S_k) = \underset{S_{k-1},S_{k-2}}{MAX}\left(\gamma_k(S_{k-2},S_k) + \alpha_{k-2}(S_{k-2})\right) \tag{13}$$

$$\beta_k(S_k) = \underset{S_{k+1},S_{k+2}}{MAX}\left(\gamma_{k+1}(S_k,S_{k+2}) + \beta_{k-2}(S_{k-2})\right) \tag{14}$$

$$\gamma_k(S_{k-2},S_k) = (\Lambda_{apr,k-1}(u_{k-1}) + y_{s(k-1)})x_{s(k-1)} + \sum_{i=1}^{m} y_{k-1}^{pi} x_{k-1}^{pi} + (\Lambda apr,k(u_k) + y_{sk})x_{sk} + \sum_{i=1}^{m} y_{pk} x_{pk} \tag{15}$$

m – number of parity bit and the sign bit of posteriori LLR value decides whether u_k=0 or u_k=1.

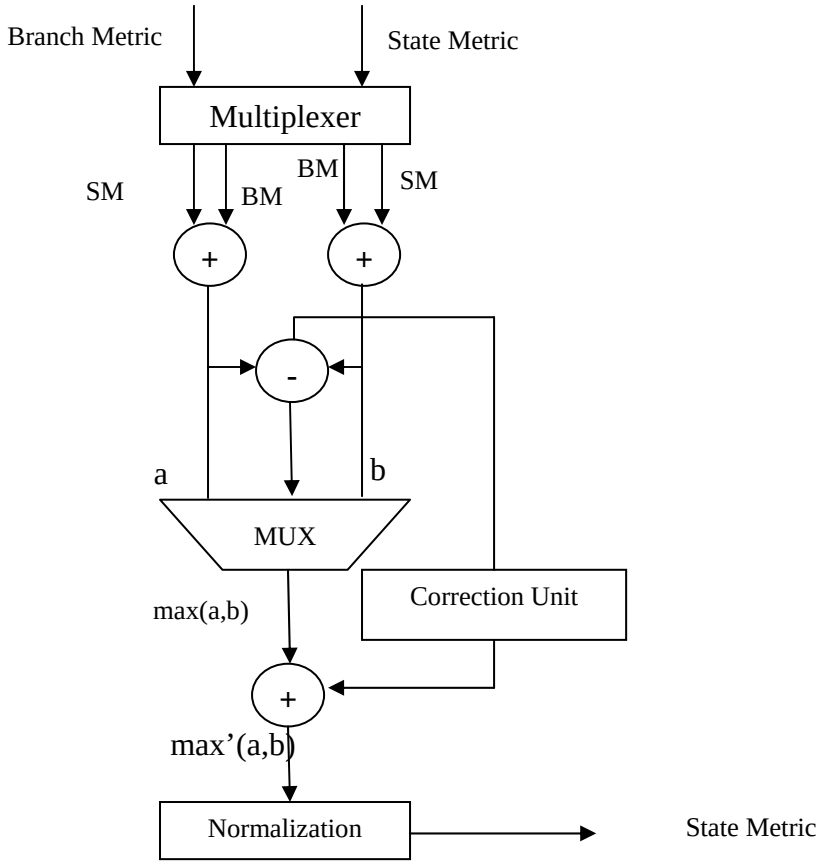


Fig. 4. Pipelined metric Architecture

ii) 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 DB MAP are described as

$$\alpha_k(S_k) = \alpha_k(S_k) = MAX_{S_{k-1}}(\gamma_k(S_{k-1}, S_k) + \alpha_{k-1}(S_{k-1})) \tag{16}$$

$$\beta_k(S_k) = MAX_{S_{k+1}}(\gamma_{k+1}(S_k, S_{k+1}) + \beta_{k+1}(S_{k+1})) \tag{17}$$

$$\gamma_k(S_{k-1}, S_k) = \Lambda^{(z)}_{apr,k}(u_k = z) + 2y_k^{s1}x_k^{s1} + 2y_k^{s2}x_k^{s2} + 2\sum_{i=1}^{m} y_k^{pi}x_k^{pi} \tag{18}$$

The reformulated equations for SB/DB are used for the calculation of Log Likelihood Ratio (LLR) calculations. By writing the MAX operations in the equations of LLR value in SB Mode, We can define $\epsilon_s(z')$

$$\epsilon_s(00) = \underset{S_{k-2}, s_k, u_{k-1=0}, u_k=0}{MAX}(\alpha_{k-2}(S_{k-2}) + \gamma_k(S_{k-2}, S_k) + \beta_k(S_k)) \tag{19}$$

$$\epsilon_s(01) = \underset{S_{k-2}, s_k, u_{k-1=0}, u_k=1}{MAX}(\alpha_{k-2}(S_{k-2}) + \gamma_k(S_{k-2}, S_k) + \beta_k(S_k)) \tag{20}$$

$$\epsilon_s(10) = \underset{S_{k-2}, s_k, u_{k-1=1}, u_k=0}{MAX}(\alpha_{k-2}(S_{k-2}) + \gamma_k(S_{k-2}, S_k) + \beta_k(S_k)) \tag{21}$$

$$\epsilon_s(11) = \underset{S_{k-2}, s_k, u_{k-1=1}, u_k=1}{MAX}(\alpha_{k-2}(S_{k-2}) + \gamma_k(S_{k-2}, S_k) + \beta_k(S_k)) \tag{22}$$

Where $z' = (u_{k-1}, u_k)$

Therefore the LLR value can be defined by

$$\Lambda(D_{k-1}(u_{k-1})) = \text{MAX}(\in_s (10),\in_s (11)) - MAX(\in_s (00),\in_s (01)) \quad (23)$$

$$\Lambda(D_k(u_k)) = \text{MAX}(\in_s (01),\in_s (11)) - MAX(\in_s (00),\in_s (10)) \quad (24)$$

By writing the MAX operations in the equations of LLR value in DB Mode, We can define $\in_d(z')$

$$\in_d (00) = \underset{S_{k-1},s_k,u_k=00}{MAX} (\alpha_{k-1}(S_{k-1}) + \gamma_k(S_{k-1},S_k) + \beta_k(S_k)) \quad (25)$$

$$\in_d (01) = \underset{S_{k-1},s_k,u_k=01}{MAX} (\alpha_{k-1}(S_{k-1}) + \gamma_k(S_{k-1},S_k) + \beta_k(S_k)) \quad (26)$$

$$\in_d (10) = \underset{S_{k-1},s_k,u_k=10}{MAX} (\alpha_{k-1}(S_{k-1}) + \gamma_k(S_{k-1},S_k) + \beta_k(S_k)) \quad (27)$$

$$\in_d (11) = \underset{S_{k-1},s_k,u_k=11}{MAX} (\alpha_{k-1}(S_{k-1}) + \gamma_k(S_{k-1},S_k) + \beta_k(S_k)) \quad (28)$$

Where $z'=(u_k)$

Therefore the LLR value can be defined by

$$\Lambda^{00}(D_k(u_k)) = 0 \quad (29)$$

$$\Lambda^{01}(D_k(u_k)) = \in_d (01) - \in_d (00) \quad (30)$$

$$\Lambda^{10}(D_k(u_k)) = \in_d (10) - \in_d (00) \quad (31)$$

$$\Lambda^{11}(D_k(u_k)) = \in_d (11) - \in_d (00) \quad (32)$$

By comparing the LLR values of both SB & DB the MAX operations of $\in s(z')$ & $\in d(z')$ are same. Therefore the hardware can be also shared. Since the hardware used for this MAX operations are shared the hardware usage and the memory used is also reduced. The power consumption is also reduced to great extent. Figure 5 shows the SB/DB Block diagram of the dual mode MAP decoder. The inputs are branch metric values and forward and backward state metric values.

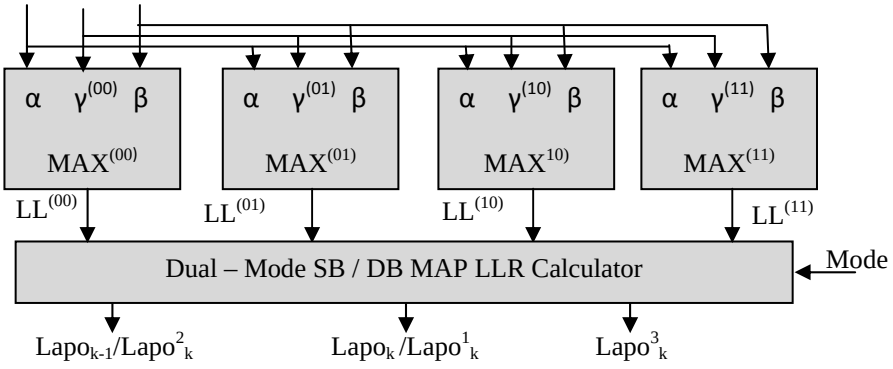
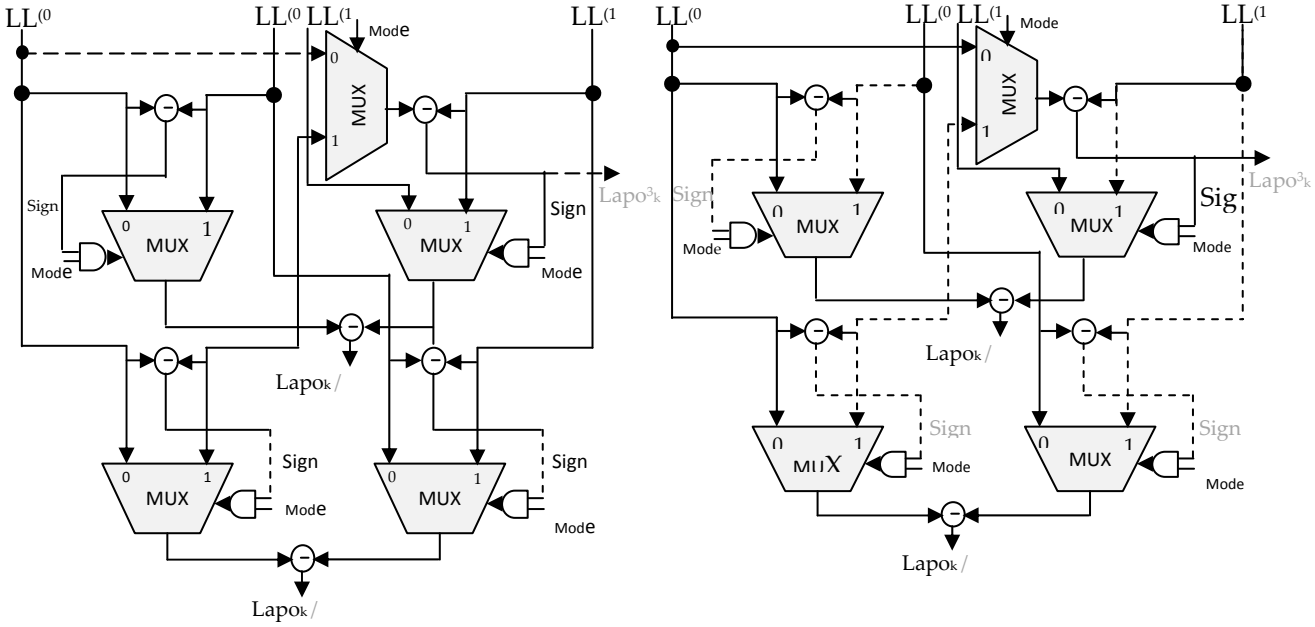


Fig. 5. Block diagram of the dual-mode MAP Decoder



(a) SB mode (b) DB mode

Fig. 6. Logic blocks of dual mode LAPO

In the Fig. 6(a) and (b) the "Mode" is used to select the mode of operation.(SB or DB). When "Mode" is active low, the dual-mode LLR calculator is in SB mode. In the dual mode architecture, some of the connections are dummy.

VI. Results and Discussion

A number of performance evaluation and resource utilization parameters are being used in the design of pipelined metric MAP decoder. The present research is focused on the design and development of pipelined metric MAP decoder for low power applications. The parameters considered for investigation include number of slices(S), number of LUTs(Look up table),Slice latches(SL),Occupied Slice latches(OSL) ,IOBs Latches(Input-Output Block),Input IBUFs (Input Buffers),gate counts(GA),Latency(L) and Power consumption(PC).

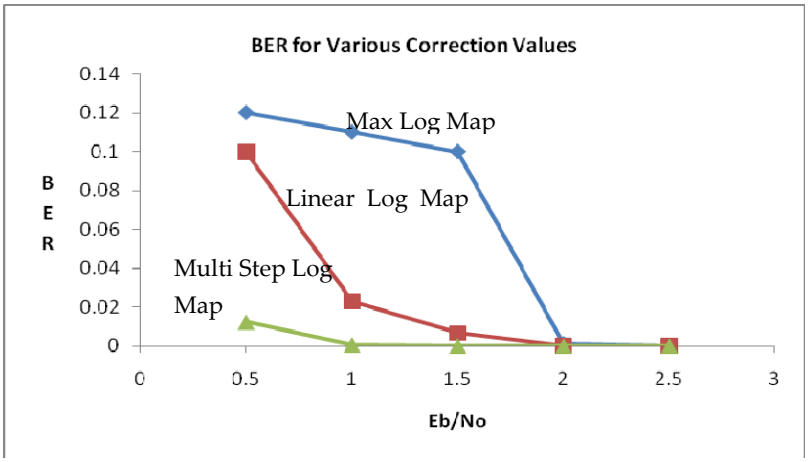


Fig. 7. Comparison of BER for Various Correction functions

The proposed decoder can be designed using Verilog HDL, a descriptive hardware language for architectural module design. This designed architectural module is simulated using Modelsim 5.5e simulating tool, synthesized using Xilinx Project Navigator 10.2i synthesis tool and tested on a Spartan3E family device XC3S500E. This proposed scheme utilized 1972 LUTs and 476 FFs at a maximum frequency of 125MHz. The proposed work results shows that the pipelined metric MAP decoding system incorporated with SB/DB mode leads to lower power consumption in terms of slices, Look Up Tables and Flip Flops.

TABLE I
Comparison of power consumptions of Spartan-3 family

FPGA Family	Device Specifications	Power Consumption
Spartan-3E	Xc3s1200E	158.9mw
Spartan-3E	Xc3s500E	81.8mw
Spartan-3E	Xc3s250E	52.27mw

The various devices in the Spartan-3 family are tested against their frequency and power and tabulated in Table 1.

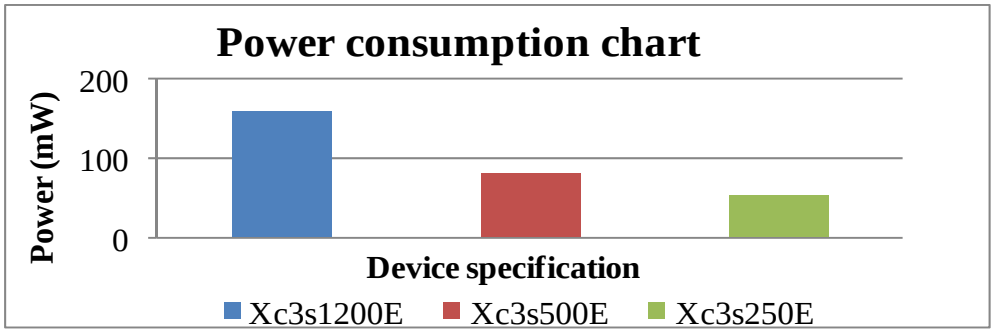


Fig. 8. Comparison of power consumptions of Spartan-3 family

Table 2 illustrates the investigation parameters, such as number of slices, LUTs, etc. required in SB and in DB mode. The values are graphically plotted and depicted in Fig. 9.

TABLE II
Performance Analysis of hardware utilization

Parameters	SB Mode	DB Mode
Slices	128	98
LUTs	72	64
Gate counts	1800	1785
IOBs	21	21

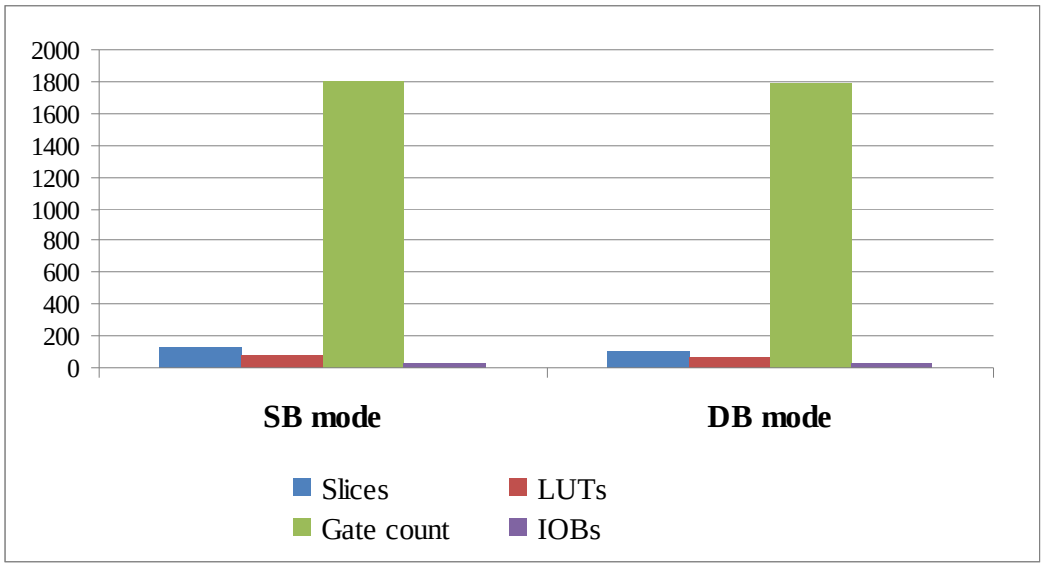


Fig. 9. Graphical representation of Hardware Utilization

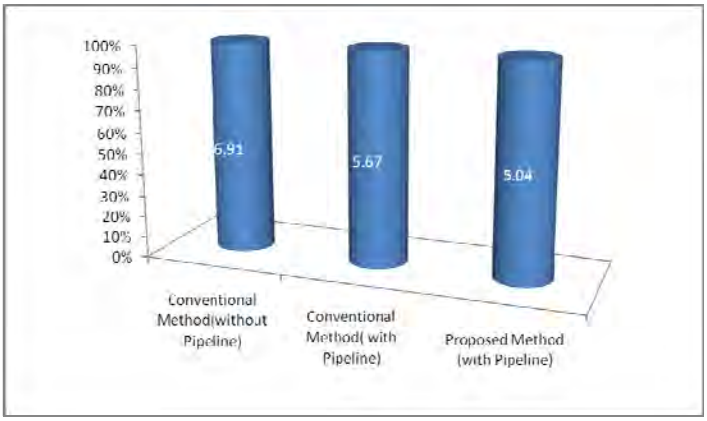


Fig. 10. Graphical representation of Power Utilization

VI. CONCLUSION

The optimum performance of Log Map Algorithm involves complex operations. By neglecting the correction function in the Max Log Map algorithm gives the capacitive loss and hence the correction term must be added which gives the optimal Log Map algorithm. The shared dual mode MAP decoder achieves low computational costs and low storages. This MAP processor achieves high throughput rates and high area efficiency.

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] C. Schurgers, F. Catthoor, and M. Engels, "Memory optimization of MAP turbo decoder algorithms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 2, pp. 305–312, Apr. 2001.
[4] 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.
[5] S.-J. Lee, N. R. Shanbhag, and A. C. Singer, "Area-efficient highthroughput MAP decoder architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 8, pp. 921–933, Aug. 2005.
[6] L.Bahl,J.Cocke,F.Jelinek and J.Raviv,"Optimal decoding of linear codes for minimizing symbol error rate." *IEEE Trans.Information Theory*, Vol.20, PP.284-287, Mar 1974.
[7] W.J.Gross and P.G.Gulak, "Simplified MAP algorithm suitable for implementation of turbo decoders" *Electronics Letters,IET Journal*, Vol 34, pp 1577-1578, Aug 1998.
[8] S.Talakoub,L.Sabeti,B.Shahrrava and M.Ahamadi,"A Linear log Map algorithm for turbo decoding and turbo equalization" *IEEE International conference on Wireless And Mobile Computing, Networking and Communications*, Vol 1, pp 182-186 Aug 2005.
[9] Hao Wang, Hongwen Yang and Dacheng Yang " Improved Log Map decoding algorithm for turbo like codes", *IEEE Communication letters*, Vol 10, pp 186-188, Mar 2006.
[10] P.Robertson,E.Vilebrun and P.Hoecher,"A Comparison of Optimal and sub optimal MAP decoding algorithms operating in log domain", *IEEE International conference on Communications*", Vol 2, pp 1009-1013, Jun 1995.
[11] Y. Sun, Y. Zhu, M. Goel, and J. R. Cavallaro, "Configurable and scalable high throughput turbo decoder architecture for multiple 4G wireless standards," in *Proc. IEEE Int. Conf. Appl.-Specific Syst., Arch. Processors (ASAP)*, 2008, pp. 209–214.
[12] C.-Y. Chen, C.-H. Lin, and A.-Y. Wu, "High-throughput dual-mode single/double binary MAP processor design for wireless WAN," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, 2008, pp. 83–87.
[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] C.-M. Wu, M.-D. Shieh, C.-H. Wu, Y.-T. Hwang, and J.-H. Chen, "VLSI architectural design tradeoffs for sliding-window Log-MAP decoders," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, Vol. 13, no. 4, pp. 439–447, Apr. 2005.
[15] Li F.-M, C.-H. Lin, and A.-Y. Wu, "Unified convolutional/turbo decoder design using tile-based timing analysis of VA/MAP kernel," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, Vol. 16, no. 10, pp. 1358–1371, Oct. 2008.
[16] T.-H. T. C.-H. Lin and A.-Y. Wu, "A memory-reduced Log-MAP kernel for turbo decoder," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2005, pp. 1032–1035.
[17] XiLinx IP Center, San Jose, CA, "*Xilinx LogiCORE, IEEE 802.16e CTC Decoder Core*," 2006.
[18] C.-H. Lin, C.-Y. Chen, A.-Y. Wu, and T.-H. Tsai, "Low-power memory-reduced traceback MAP decoding for double-binary Convolutional turbo decoder," *IEEE Trans. Circuits Syst. I, Reg. Papers*, Vol. 56, No. 5, pp. 1005–1016, May 2009.
[19] C.-H. Tang, C.-C. Wong, C.-L. Chen, C.-C. Lin, and H.-C. Chang, "A 952MS/s max-log MAP decoder chip using radix-4 4 ACS architecture," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, 2006, pp. 79–82.
[20] M.-C. Shin and I.-C. Park, "Processor-based turbo interleaver for multiple third-generation wireless standards," *IEEE Commun. Lett.*, Vol. 7, no. 5, pp. 210–212, May 2003.