



Research Journal of
**Information
Technology**

ISSN 1815-7432



Academic
Journals Inc.

www.academicjournals.com

FPGA Implementation of Turbo Decoder for IDMA Scheme

¹Kuttathatti Srinivasan Vishvaksenan, ²R. Seshasayanan and ¹D.V. Hariprasad

¹Department of Electronics and Communication Engineering, SSN College of Engineering, Chennai, India

²Department of Electronics and Communication Engineering, Anna University, India

Corresponding Author: Kuttathatti Srinivasan Vishvaksenan, Department of Electronics and Communication Engineering, SSN College of Engineering, Chennai, India

ABSTRACT

In this study, we considered proposed multiple accessing scheme called IDMA receiver. An iterative decoding algorithm (turbo decoder) is used in IDMA receiver. Turbo code becomes standard encoder for the future mobile communication system. Different high-speed architectures for SISO block of a turbo decoder are explored and a structure based on a new SISO is proposed. The new high-speed and low-complexity architecture serves high quality turbo-coding applications. According to the system-level simulation and architectural explorations on the area and speed of the implemented SISO block, the optimized size of internal memories and logic cells are determined. The architecture is described using Verilog modeling and synthesized for xilinx's FPGA platforms. The new SISO block is verified and compared with the various previously presented SISO blocks. The obtained results of synthesis for hardware area and speed complexities are presented to satisfy the proposed architecture design capabilities.

Key words: Soft Input Soft Output (Siso), Field Programmable Gate Array (FPGA), Maximum Likelihood-Maximum a Posteriori Probability (ML-MAP), Log-Likelihood Ratio (LLR)

INTRODUCTION

Interleave-Division Multiple Access (IDMA) is a multiple access scheme in which different interleavers are used to distinguish users as against different codes in a conventional Code-Division Multiple-Access (CDMA) scheme. The basic principle of IDMA is that the interleaver is unique for the users. Interleavers are generated independently and randomly. Turbo Codes (TC) are one of the most powerful forward error correction channel codes. IDMA scheme in conjunction with TC can approach the Shannon bound by a decibel or less with iteratively processed maximum-a posteriori decoders (Emmanuel *et al.*, 2007). Because of its outstanding performance, rich applications can be found in 3, 3.5 and 4G standards, such as WCDMA, WiMax, LTE, IDMA Ping *et al.* (2006) accessing scheme etc. It is well known that MAP (Maximum Posteriori Probability) results in less probability of error. It maximizes the probability of taking correcting decisions. The recursive computations inherent with the log-MAP algorithm (Proakis, 2006) are the main issues, which limit the throughput of the decoder. Several high-throughput architectures of Log-MAP decoder have been presented recently: The Lucent Bell Labs firstly presented a novel radix-4 architecture 3GPP TS 25.212 (2002), which can generally improve the processing speed by incorporating two stages of state transitions together, while suffering from a performance loss of 0.04 dB because of the approximation (Mettas *et al.*, 2000). Zhang and Jiang (2008) presented a high-speed radix-2 architecture, which showed nearly the same performance

as radix-4 architecture proposed in 3GPP TS 25.212 (2002). Al-Mohandes and Elmasry (2002), adopted the Block-Interleaved Pipelining (BIP) technology at the architectural level to design a high-throughput decoder. Robertson *et al.* (1995), used two Look-Up Tables (LUTs) to improve the accuracy at the cost of an extra add operation and therefore a hardware increase. Al-Mohandes and Elmasry (2003), Moreno and Aguilar (2003) and Yeo *et al.* (2003), introduced four subtractors with one combinational logic cell to improve the speed at the cost of hardware increase.

In this study, we propose an efficient VLSI architecture for turbo decoding. This architecture can be Configured to support both simple and double binary turbo codes up to eight states. We address the memory collision problems by applying new highly parallel interleavers. The MAP decoder (Proakis, 2006), memory structure and routing network are designed to operate at full speed with the parallel interleaver. The proposed architecture meets the challenge of multi-standard turbo decoding at very high data rates.

TRANSMITTER AND RECEIVER STRUCTURES FOR CDMA AND IDMA

In Fig. 2, a DS-CDMA system is illustrated where the arrangement of interleaving and spreading is reversed. Now, different interleavers distinguish distinct data streams. This special case of DS-CDMA is called code-spread CDMA, chip-interleaved CDMA (cI-CDMA), or Interleave-Division Multiple Access (IDMA) in the literature.

In IDMA, a single low-rate encoder, subsequently denoted by ENC, may do FEC encoding and spreading jointly. The spreader has no special function. Furthermore, it is important to note that interleaving is done on a chip-by-chip basis.

Figure 3 show Transmitter and (iterative) receiver structures of an IDMA scheme with K simultaneous users.

MAP decoding algorithm: The turbo decoding concept is functionally illustrated in Fig. 1. The decoding algorithm is called the Maximum a Posteriori (MAP) algorithm and is usually calculated in the log domain (Al-Mohandes and Elmasry, 2002; Mazza *et al.*, 2002). During the decoding process, each SISO decoder receives the intrinsic Log-Likelihood Ratios (LLRs). From the channel and the extrinsic LLRs from the other constituent SISO decoder through interleaving (II) or deinterleaving (II⁻¹) Proakis (2006). Consider a decoding process of simple binary turbo codes, let s_k be the trellis state at time k, then the MAP decoder computes the LLR of the A Posteriori Probability (APP) of each information bit u_k by:

$$\Lambda(\hat{u}_k) = \max_{U: u_k=1}^* \{ \alpha_{k-1}(s_{k-1}) + \gamma_k(s_{k-1}, s_k) + \beta_k(s_k) \} \\ - \max_{U: u_k=0}^* \{ \alpha_{k-1}(s_{k-1}) + \gamma_k(s_{k-1}, s_k) + \beta_k(s_k) \}$$

where, k+1 and k-1 represent forward and backward state metrics, respectively and are computed recursively as:

$$\alpha_k(s_k) = \max_{s_{k-1}}^* \{ \alpha_{k-1}(s_{k-1}) + \gamma_k(s_{k-1}, s_k) \} \quad (1)$$

$$\beta_k(s_k) = \max_{s_{k+1}}^* \{ \beta_{k+1}(s_{k+1}) + \gamma_k(s_k, s_{k+1}) \} \quad (2)$$

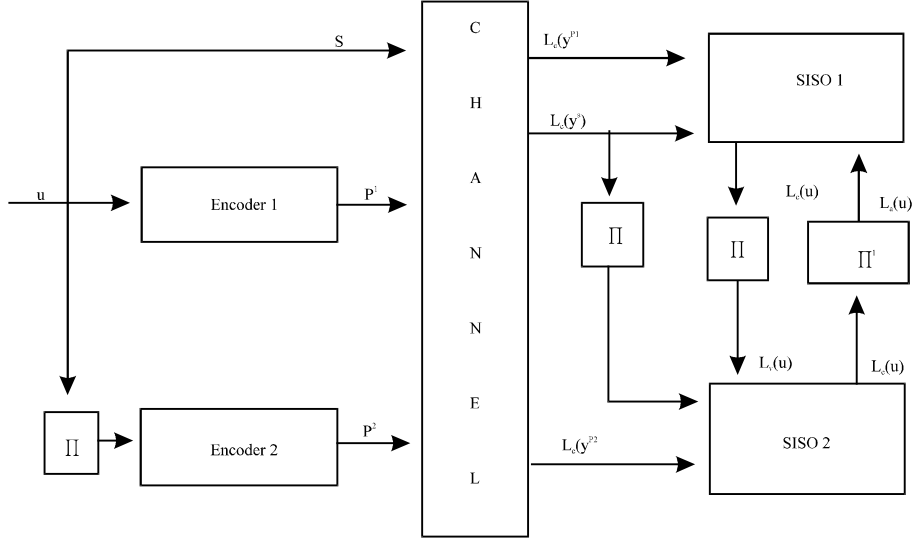


Fig. 1: Turbo encoder/decoder structure

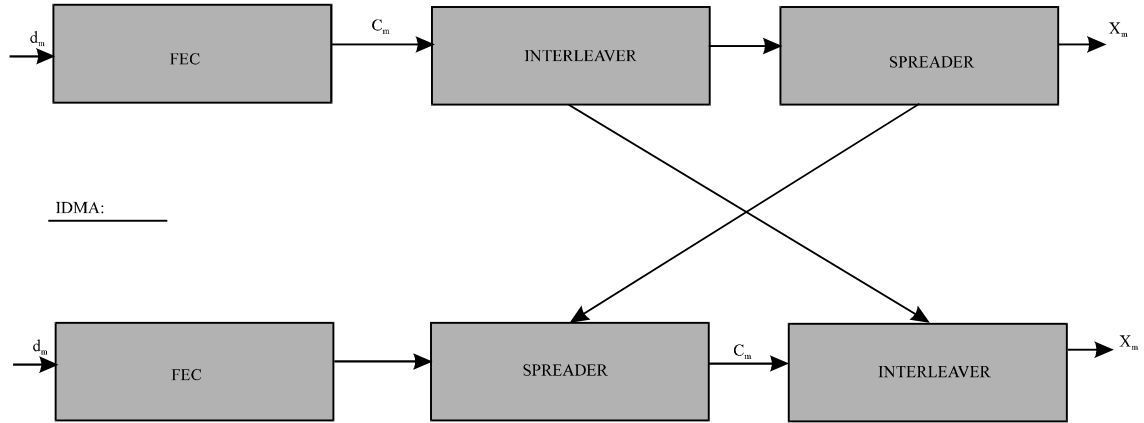


Fig. 2: Comparison of conventional DS-CDMA with IDMA. The data sequence of the m -th layer is denoted as d_m and the corresponding encoded and interleaved sequence is denoted as x_m

The second term k is the branch transition probability and is usually referred to as a Branch Metric (BM) (Proakis, 2006). To extract the extrinsic information $Ex_{\hat{u}}(k)$ is split into three terms: extrinsic information $L_e(\hat{u}(k))$, a priori information $L_a()$ and systematic information $L_c(\hat{u}(k))$ as:

$$Ex(\hat{u}(k)) = L_e(\hat{u}(k)) + L_c(ysk)$$

DECODING PROCESS USING SLIDING WINDOW

The conventional MAP decoding process has very high latency due to the processing of forward and backward calculations in all trellis states. Computing the LLR values requires the state metric values generated by the forward and backward processes. Therefore, a large memory size is required to store the state metric values, which in turn depends on the input data block size. The

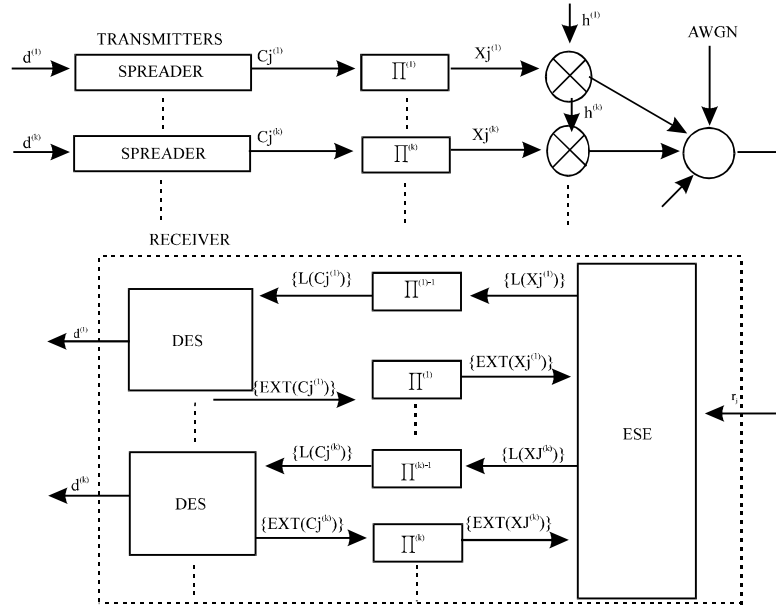


Fig. 3: IDMA transmitter and receiver

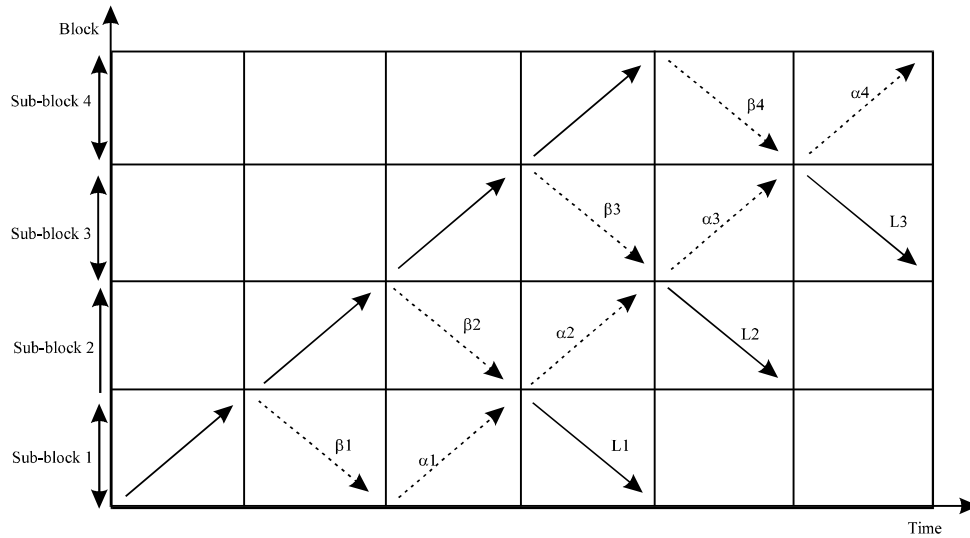


Fig. 4: A graph of data flow with sliding window method

Sliding Window method is used to reduce the memory size by dividing the input data into sub-blocks. In this architecture, the sub-block size is fixed and it can also have variable block sizes. In this proposed design the smallest UMTS standard specification is taken to fix the sub-block size as 40. The Fig. 4 shows the graph of data flow for decoding process in time and block axis.

ML-MAP SISO DECODER ARCHITECTURE

Figure 5 shows the SISO decoder architecture, which consists of the forward and backward state metric, LLR computation and memory (LIFO and FIFO) blocks. The FIFO 1 and 2 are used to Buffer the input data symbols and the LIFO 3 and 4 are to store the forward state metric and the LLR values, respectively.

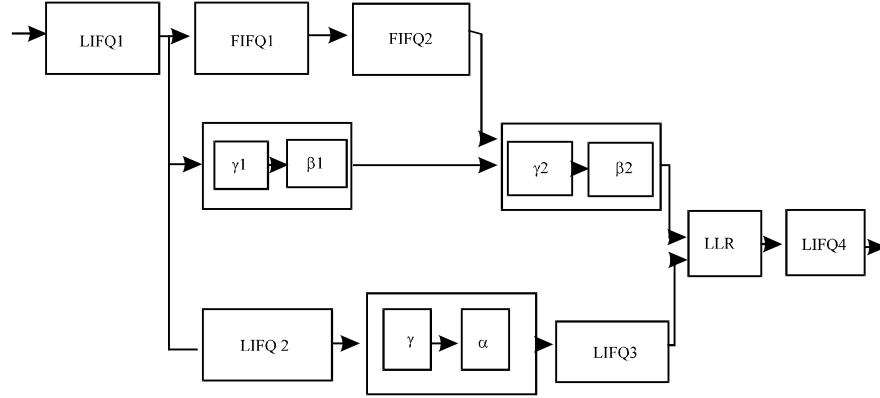


Fig. 5: SISO architecture

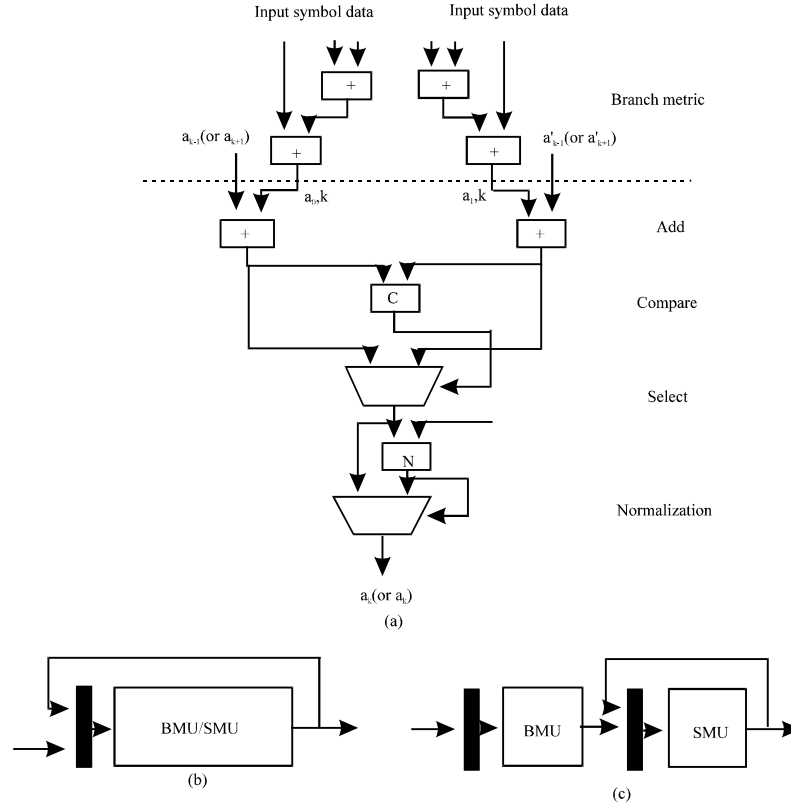


Fig. 6: Conventional structures of the Branch Metric Unit (BMU) and State Metric Units (SMU)

The SISO decoder has been built with two backward state metric units, β_1 and β_2 . α and γ denote the forward state and branch metric units.

BRANCH AND STATE METRIC UNIT

The Branch and State Metric Units (BMU and SMU) are implemented using ML-MAP algorithm. The conventional BMU and SMU consists of branch metric calculation, add, compare, select and normalization Processes, which are shown in Fig. 6. The general SMU in turbo SISO

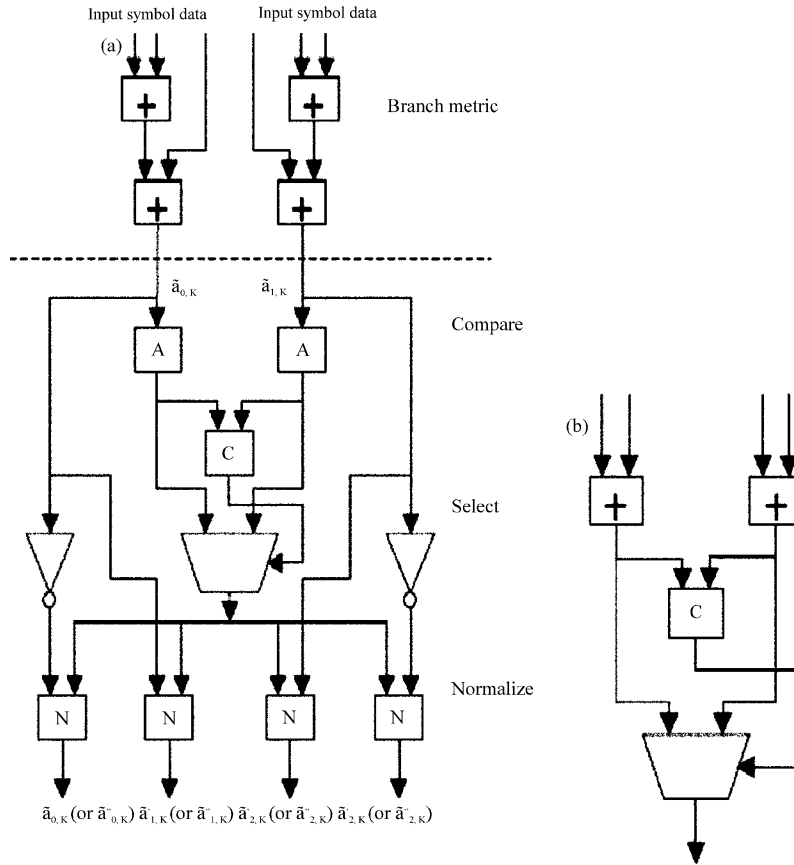


Fig. 7: The proposed structure of (a) BMU and (b) SMU

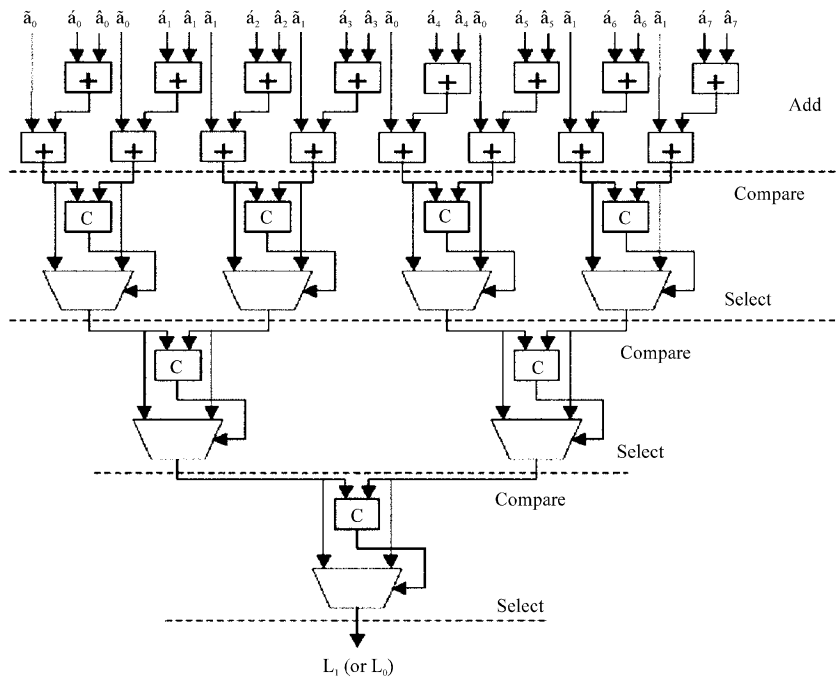


Fig. 8: Conventional LCU units

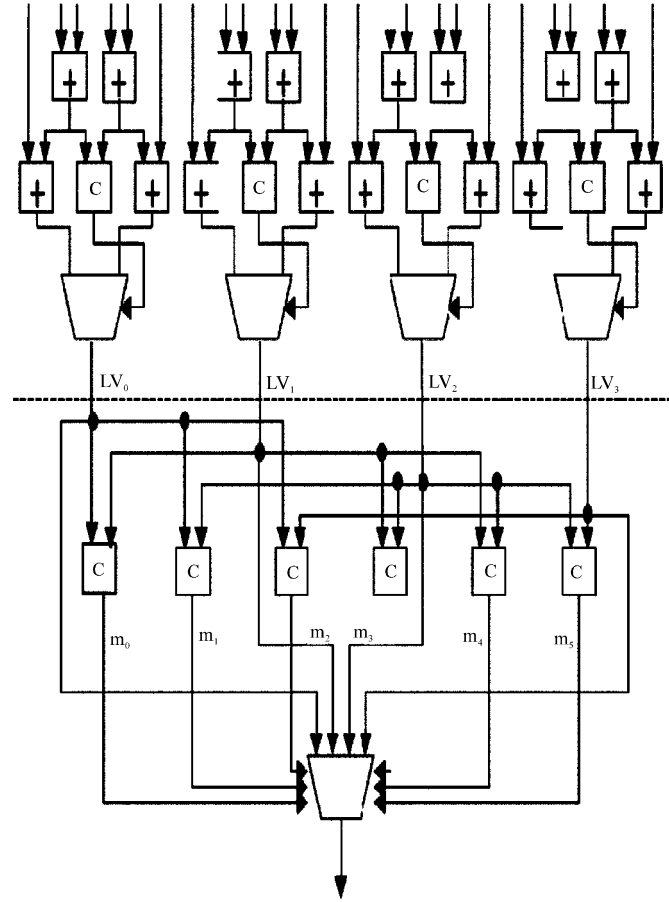


Fig. 9: Proposed LCU structure

decoder must include the normalization process to avoid overflow of the state metric values. The branch metric values are obtained from input data symbols. The new state metric values are calculated in single clock cycle recursively using add, compare (C), select and normalization (N) processes from branch metric and state metric values. The above processes determine the critical path delay.

In the proposed architecture, which is shown in Fig. 7a, the normalization is done in the branch metric values itself. This normalization method leads to a simplified SMU, but more complex in BMU. The novel architecture reduces the critical path delay significantly by eliminating the state metric normalization process used in the conventional SMU.

LLR COMPUTATION UNIT

The LLR values (L_1 or L_0) are calculated using forward ($\alpha_{0..7}$) and backward ($\beta_{0..7}$) states and branch metric ($\gamma_{0..1}$) values of all states. The LLR computation unit (LCU) is similar to the SMU, which consists of 3-stage compare and select process results long critical path delay. In order to reduce the critical path delay LCU is pipelined.

SIMULATION RESULTS

The turbo decoder architecture is proposed and implemented using Verilog HDL, the simulation for turbo decoder is done by modelsim and synthesis by Xilinx. The Fig. 10 and 11 shows the simulation and synthesis result of SISO Decoder. The Table 1 shows the Device utilization summary of SISO decoder. Figure 12 elucidate that turbo decoder can achieve Bit error rate close to 10^{-5} with less SNR.

Fig. 10: SISO Decoder

Table 1: Device utilization selected device: 3s500eft256-4

Description	Used	Total	Percentage
No. of slices	184	4656	3
No. of slice Flip Flops	162	9312	1
No. of 4 input LUTs	311	9312	3
No. of bonded IOBs	33	190	17
No. of GCLKs	1	24	4

Minimum period: 8.056 ns Maximum frequency: 124.130 MHz

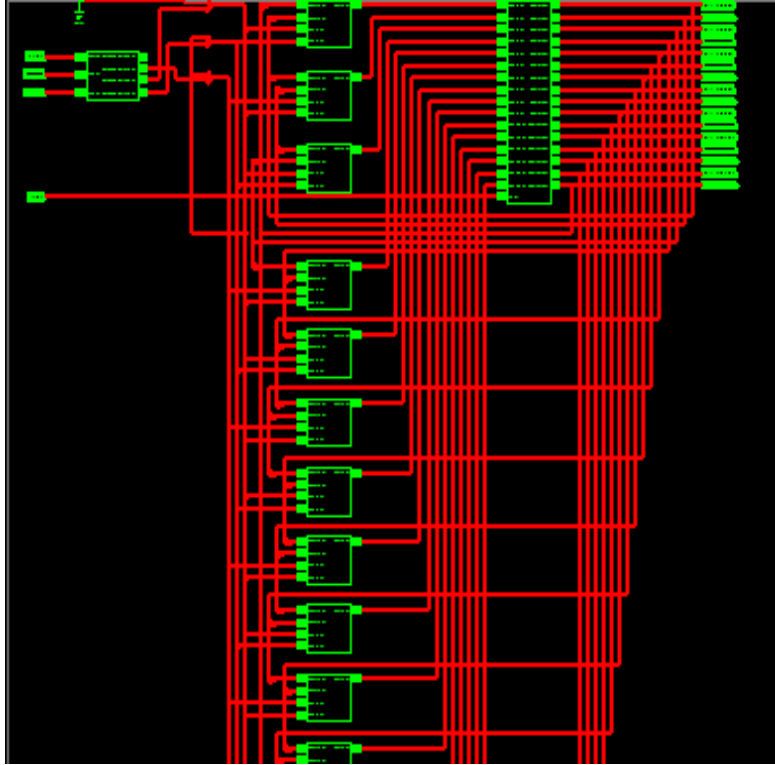


Fig. 11: RTL schematic of turbo decoder

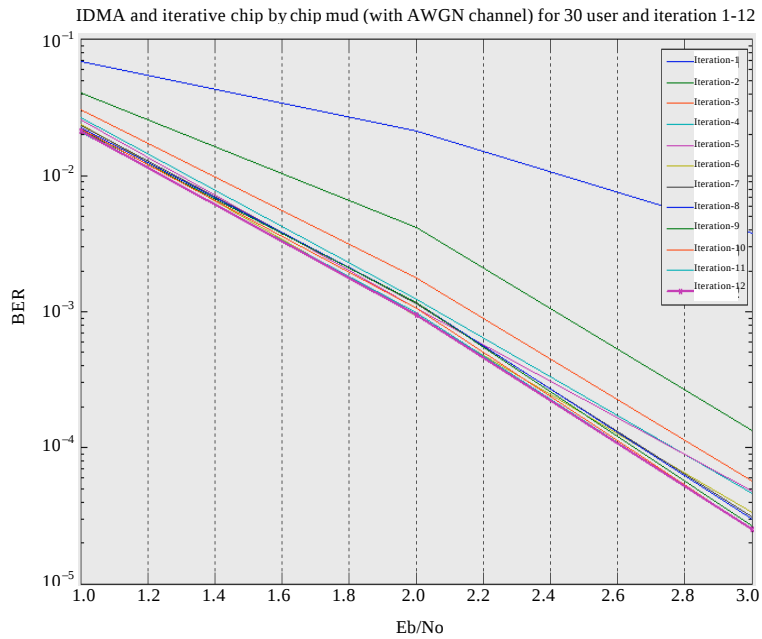


Fig. 12: The performance of iterative turbo decoder for IDMA with MUD for 30 user and iteration-12

CONCLUSION AND FUTURE WORK

In this study, different high-speed architectures for SISO block of a turbo decoder have been considered and a structure based on a new SISO is proposed. In future wireless mobile communication system, turbo code is prescribed as channel encoder. Iterative decoding algorithm is used in turbo code to produce BER of 10^{-5} . The simulation results evince that IDMA in conjunction with turbo code is supporting many users. Therefore high speed and low-complexity architecture is essential to meet our requirement such as high data rate etc., According to the system-level simulation and architectural explorations on the area and speed of the implemented SISO block, the optimized size of internal memories and logic cells are determined. The architecture is described using Verilog modeling and synthesized for xilinx's FPGA platforms. The results show that the architecture operates at a maximum frequency of 124 MHz.

REFERENCES

- 3GPP TS 25.212, 2002. Multiplexing and channel coding (FDD). Version 4.6.0 Available 2002-09-27.
- Al-Mohandes, I.A. and M.I. Elmasry, 2002. A new efficient dynamic-iterative technique for turbo decoders. Proceedings of the 45th Midwest Symposium on Circuits and Systems, Aug. 4-7, USA., pp: 180-183.
- Al-Mohandes, I.A. and M.I. Elmasry, 2003. Design of energy-efficient turbo decoder for 3RD wireless applications. Proceedings of the 15th International Conference, Dec. 9-11, VLSI Res. Group, Waterloo Univ., Ont., Canada, pp: 127-130.
- Emmanuel, B., C. Douillard and G. Montorsi, 2007. Iterative decoding of concatenated convolutional codes: Implementation issues. Proc. IEEE, 95: 1201-1227.
- Mazza, M., M. Piccinini, G.F. Viglione and M. Zamboni, 2002. Architecture strategies for low-power VLSI turbo decoders. IEEE Tran. VLSI Syst., 10: 279-285.
- Mettas, A., ReliaSoft Corporation and Tucson, 2000. Reliability allocation and optimization for complex systems. Proceedings of the Annual Reliability and Maintainability Symposium, Jan. 24-27, Los Angeles, California, USA., pp: 1-7.
- Moreno, R.S. and A.R. Aguilar, 2003. Failure Distribution at Component Level. American Society of Agricultural and Biological Engineers, St. Joseph, Michigan.
- Ping, L., L. Liu, K. Wu and W.K. Leung, 2006. Interleave division multiple-access. IEEE Trans. Wireless Commun., 5: 938-947.
- Proakis, J.G., 2006. Fundamentals of Communication Systems. Pearson Education, Prentice Hall, USA.
- Robertson, P., E. Villebrun and P. Hoeher, 1995. A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain. Proceedings of the International Conference on Communications, June 18-22, Seattle, WA. USA., pp: 1009-1013.
- Yeo, E., B. Nikolic and V. Anantharam, 2003. Iterative decoder architectures. IEEE Commun. Magazine, 41: 132-140.
- Zhang, Y. and J. Jiang, 2008. Bibliographical review on reconfigurable fault-Tolerant control systems. Ann. Rev. Control, 32: 229-252.