Pak. J. Biotechnol. Vol. 13 (special issue on Innovations in information Embedded and Communication Systems)
Pp. 242- 246 (2016)

# RECONFIGURABLE MULTIPROCESSOR ARCHITECTURE FOR TURBO DECODING

[1]R.Ilakkiya and [2]A.Kalaiselvi

Department of ECE Kumaraguru College of Technology Coimbatore, Tamilnadu
[1]rilakkiya39@gmail.com; [2]kalaiselvi.a.ece@kct.ac.in

## ABSTRACT

This paper presents the implementation of turbo decoder on butterfly topology and star topology for parallel processing. Now a days turbo decoder plays major role in wireless communication like WIFI, WIMAX etc. Turbo codes are replacing the LPDC code due to it's higher error correcting ratio. Turbo codes are generated by the turbo encoders in the transmitter side. During transmission turbo codes are corrupted with noise. The corrupted signal is reconstructed using turbo decoder. The butterfly topology are implemented in turbo decoder for asynchronous load. It provides less network latency and less path diversity. The star topology is implemented in turbo decoder for simple architecture. Turbo decoder implemented in butterfly topology and star topology produce high throughput compared to normal turbo decoding.

*Index Terms*— Turbo Decoder, Butterfly Topology, Performance analysis

## INTRODUCTION

Channel codes are used to protect digital data from errors by introducing redundancies in the transmitted data. Channel codes that are used to detect errors called error detection codes and which are used to correct errors are called error correction codes. Among many error correcting codes turbo codes are new class of error correcting codes which have been proved to provide a considerably better performance than existing traditional coding schemes [1]. It combines the capabilities of convolutional codes with channel estimation theory. It has high coding gain when compared to existing error correcting codes. It requires receiver that can determine the instantaneous SNR of the link. The main application of turbo codes are frequency concentrated in the field of communication specially mobile communications [2]. Main characteristics of turbo codes are: Use of soft information. The signal which is detected from the channel, the turbo decoder expects only the soft information not the hard information [2]. In other words, turbo decoders have received symbol like statistics or probability information which is close to 0 or 1 and processing the information in determining the hard decision later.

Channel coding technique is associated with many parameters like frame size, communication channel, signal to noise ratio. Among channel coding techniques, turbo codes (TCs) are frequently adopted in the recent wireless standards. Turbo codes are frequently used because it achieves very low bit error rate BER [1]. In recent wireless communication standards, such as WiMax and LTE, the use of contention free interleavers enables to achieve high throughput implementtations presented in [3] and [8]. These architectures propose the use of multiple Soft-Input Soft-Output (SISO) decoders enables to reach the high-throughput requirement of emerging and future standards. Among channel coding techniques turbo decoders offer certain degrees of flexibility. SISO decoder solutions have been investigated in order offer architectures providing good tradeoff in terms of flexibility, throughput and power dissipation [1].

**Overview of Turbo Decoding using SISO Decoder**

Turbo decoder is one of the most difficult blocks in any communication channel which requires high throughput, adequate area and low power [1]. The efficient implementation of turbo decoder is to improve the overall system performance.

*A. Turbo Encoder Structure*

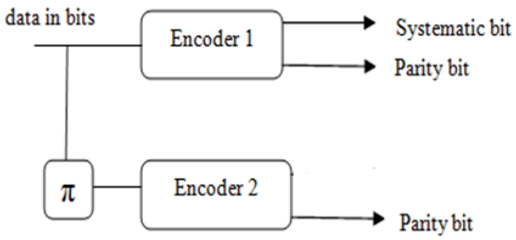


Fig.1.Encoder Structure

At the transmitter side, it consists of two encoders [3]. Data in bits is given as input to the encoder 1 which produces two outputs like systematic bit and parity bit. And the same input is interleaved before it passes to the encoder 2 which produces interleaved bit and parity bit. Both the encoders work in the similar way. It is also called 4-state Finite State Machine, because it generates a parity bit for each received bit. So totally we have four sequences i.e original sequence, parity bit from system 1, then interleaved sequence, and parity bit from system 2. We can send any type of information sequence. Depending on bandwidth, we chose how much information we want to send. The decoder can work with any type of information sequence. Then the sequence is modulated and then transmitted over the channel. Encoded output data bits are passed through the channel. Some of data bits get corrupted due to noises and error may be introduced in the system [3]. The corrupted data bits are transferred to the decoder part where the error is removed and original data is recovered.

*B. Turbo Decoder Structure*

Turbo decoder consist a two SISO decoder. Turbo decoders works is based on an exchange of statistics or probabilistic information known as extrinsic informa-

tion between some component decoders but it deals with the same received set of information [4].

At the decoder:

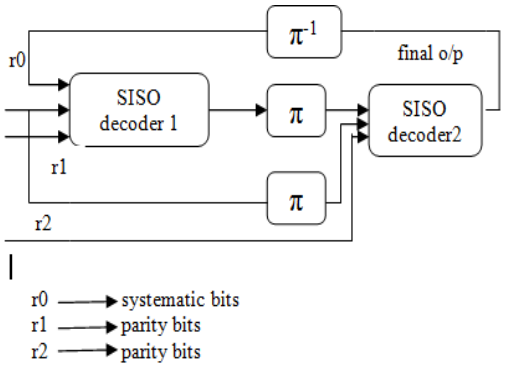


Fig 2: Decoder Structure.

The encoded data is then transmitted through the channel and then reaches the SISO decoders. The original data and the parity bit of encoder 1 is given to the SISO decoder 1. The output of SISO decoder 1 gives the extrinsic information. That extrinsic information of decoder 1 is given as input to the decoder 2. Then the original data is interleaved and given to the SISO decoder 2. Then the parity bit of encoder 2 is given to the SISO decoder 2. So the output come from the decoder 2 is the final output. To improve the performance of turbo codes, the final output is deinterleaved and then gives again as input to the decoder 1.

Both SISO's processing the sequence and exchange the information until the soft information matches to acceptable range. From the final soft information sequence the hard decision will be made later. In order to improve its estimation over the iterations, each SISO decoder deals with the extrinsic information generated by the other SISO decoder. The computations are done in the logarithmic domain usually, but not necessary.

*C. Purpose of Interleaver*

Interleaver unit is a random block which is used to rearrange the input data bits with no repetition. Interleaver unit is used in encoder part and as well as decoder part. It generates a long block of data, at the encoder side [4]. It correlates the two decoders and helps to correct the error in decoder part. After passing the encoded data to the first decoder only some of the errors may get corrected, Again we interleaved the corrected data, and then given to the second decoder. Then, remaining error get corrected in the second decoder [5]. In this way, we are repeating the process for large number of times.

*D. Decoding Algorithms*

Decoding algorithms can be calculated by using a posteriori probability method or maximum likelihood decoding method [2]. Each decoder calculates the Log-Likelihood Ratio (LLR) for the $i^{th}$ data bit $d_i$ as

$$LLR = \ln \frac{P(u_l = 0, r)}{P(u_l = 0,1)}$$

$$P(u_l = 0, r) = \sum_{(s',s)} P(s_l = s', s_{l+1} = s, r)$$

$$P(s_l = s', s_{l+1} = s, r) = \alpha_{l-1}(s')\gamma_l(s', s)\beta_l(s)$$

$$\alpha_{l-1}(s') = P(s_l = s', r_0^{l-1})$$

$$\gamma_l(s', s) = P(s_l = s, r_l^{(1)}, r_l^{(2)} / s_l = s')$$

$$\beta_l(s) = P(r_{l+1}^{(k+1)} / s_{l+1} = s)$$

where α, β and γ are the forward recursion, backward recursion and branch metrics respectively. LLR value for each received bit can be calculated by multiplying the forward recursion, backward recursion and branch metrics[5]. Usually LLR values are calculated from the MAP algorithm. If LLR value of each received bit is positive, then the bit is mostly said to be '1'and if value is negative then the bit is said to be '0'. The max log map algorithm is given as

- It is symbol by symbol detection
- Derived from log map algorithm
- In summation of probabilities in Log-Map algorithm , we are using max*(.) operation
- The max*(.) operation requires to convert LLR value into an exponential form
- Because at each state in forward and backward calculations only the path with maximum value is considered
- The probabilities are not calculated over all the code words.
- So, Max Log MAP algorithm is least complex.

*E. Maximum Likelihood Decoding Method*

Decoding is done mostly using maximum likelihood decoding method [5]. It is given by P (x received/y sent) =P(x received, y sent)/P(y sent)

It is also given as

$$P(A / B) = \frac{P(A, B)}{P(B)}$$

$$P(A, B) = P(B).P(A / B)$$

*F. Implementation of Turbo Decoder Using Butterfly Topology*

The received data transfers through the butterfly NoC. NoC- Network on Chip. It is a communication subsystem on an IC which is called chip. The Butterfly NoC is a multistage interconnection network. It allows a different path in the topology between each pair of nodes i.e., (from source to destination)[6]. In this topology there are two inputs and two outputs . From the fig 3. it is clear that when data's are received in BT1 and BT2 it splits the data into two form. One type of data is stored in BT5 through straight forward method. It act as a first input in BT5. Another type of

data from BT1 and BT2 is get subtracted and stored in BT5 as second input. Then again data from BT1 and BT2 is get added and stored in BT6 as first input. Then another data from BT2 is stored in BT6 through straight forward method. This calculation is proceed for further decoders respectively. After this calculation , shuffled operation will takes place. After that all the outputs are stored in respective receivers.

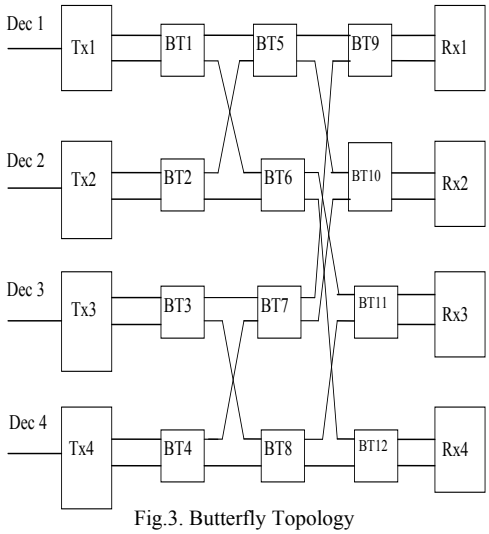


Fig.3. Butterfly Topology

Finally a single bit is used which is used to select the required output: 1) 0 for the first output and 2) 1 for the second output. This topology will allow each turbo decoder to connect to a different registers so that only all turbo decoders are active without any disruptions. In this topology there is an interior block which is called as a switch. That switch is configured to pass the data through straight forward like top or to the opposite side like bottom. The butterfly network structure can handle multiple turbo decoders, so it has good performance. The butterfly topology structure has advantages of reducing network latency and so it performs well. It does not have path diversity. This topology is mostly preferred due to asynchronous loads. When heavy load is given as input, butterfly topology is used. Due to this method, complicated situation will be avoided.

*G.Implementation of Turbo Decoder Using Star Topology*

Star topology is one of the most common network topologies. It's structure is simplest in nature, which consists of one central hub. It acts as a conduit to transmit messages. That central hub is said to be configuration manager, to which all other turbo decoders are connected. In star topology, each turbo decoder connected to a configuration manager using a point-to-point connection. . The start up costs are low and so it becomes very popular. In star topology, it is also easy to add more turbo decoders to the topology. This topology is robust in the sense that if one connection between a turbo decoder and the configuration manager fails, then the other connections remain active. It requires more cable than the other topologies and also more expensive. It decreases the damage caused by line failure by joining all of the turbo decoders to a configuration manager. The star topology becomes very popular because it is easy to wire and install. Then there is no disruptions to the topology when connecting or removing further turbo decoders. Finally it is easy to detect the faults occurring in which turbo decoder and remove that parts. Star topology is also said to be reliable because if one turbo decoder or its connection breaks it doesn't affect the other turbo decoders and their connections.

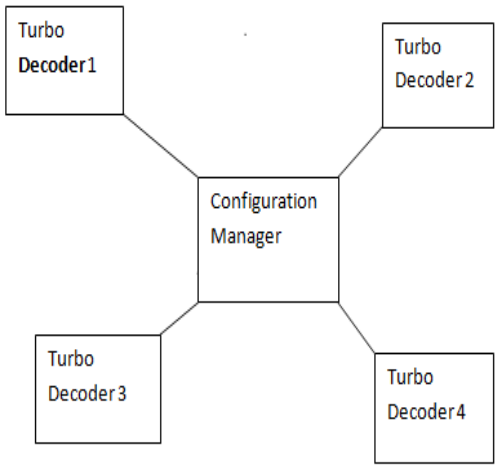


Fig 4: Star Topology

## SIMULATION RESULTS

Turbo decoders are implemented using butterfly topology and star topology in VHDL for standard bit i.e. 24 bit. A reset signal is given first to clear the register and output will be zero for reset operation. The decoder output will be produced for subsequent clock cycle. MAX-LOG-MAP algorithms are used to update the value to get output equivalent to the desired signal. Then the frame size is kept constant and increases the number of decoders. By increasing the number of decoders and keeping the frame size as constant we are going to calculate the two parameters like transfer latency in terms of nano seconds and number of input LUT's. Calculate these two parameters for two topologies (butterfly and star topology). The performance of turbo decoder using butterfly topology is given below:

TABLE -I:. COMPARISON OF VARIOUS DECODERS USING BUTTERFLY TOPOLOGY

| No. of decoders | Transfer latency (in ns) | No. of input LUTs |
|---|---|---|
| 4 | 6.322 | 25 |
| 8 | 6.016 | 28 |
| 16 | 5.842 | 33 |

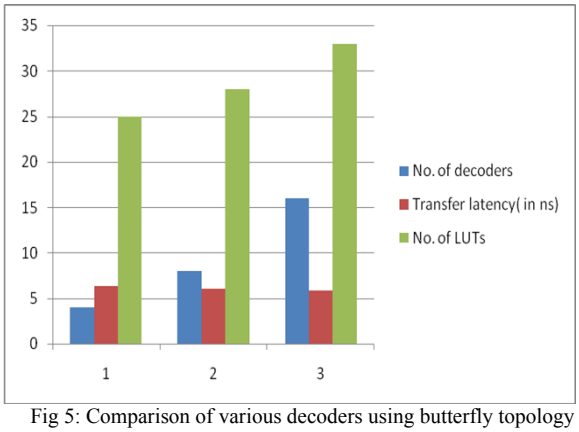


Fig 5: Comparison of various decoders using butterfly topology

.

TABLE II. COMPARISON OF VARIOUS DECODERS USING STAR TOPOLOGY

| No. of decoders | Transfer latency (in ns) | No. of input LUTs |
|---|---|---|
| 4 | 6.478 | 24 |
| 8 | 6.121 | 26 |
| 16 | 5.735 | 31 |

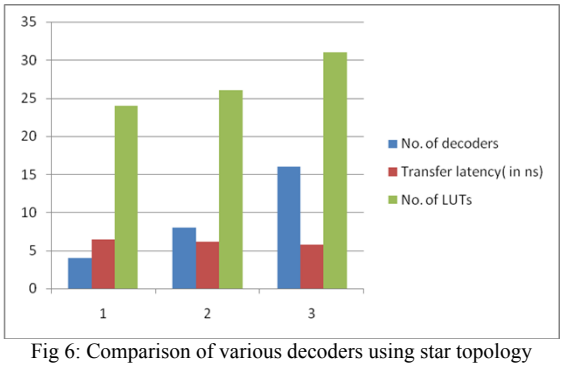


Fig 6: Comparison of various decoders using star topology

## *A. RTL SCHEMATICS*

The RTL Schematics of using 4 turbo decoder is given in figure 7. The input signal is represented by data, clk and reset. The output signal of turbo decoder is represented by dataout, d1chpart1, d2chpart2, d1chpart3, d1chpart4 and errout.

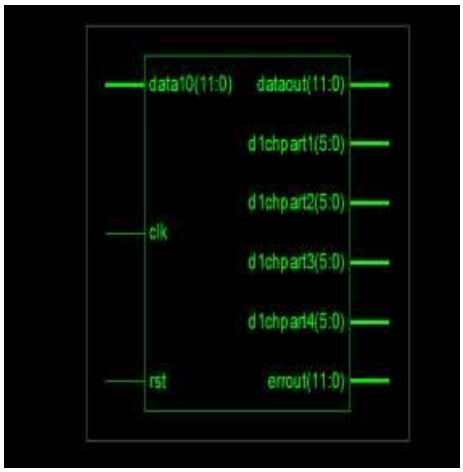


Fig 7. RTL schematic of turbo decoder

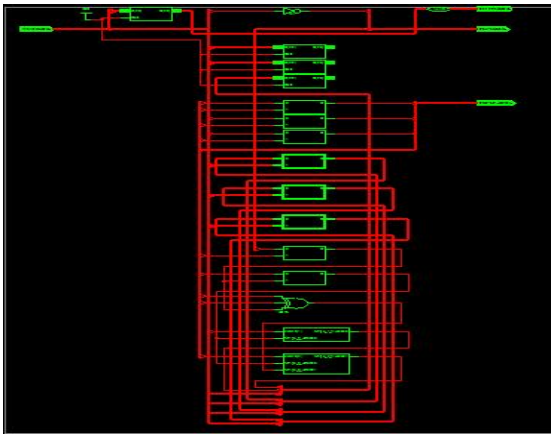

Fig 8: Sample RTL schematic diagram.

## Conclusion

The turbo decoder is implemented for parallel processing to increase the speed of the system. Butterfly topology is developed for asynchronous load. It also developed for higher order bits to verify it's performance. Further the Star topology can be implemented in the turbo decoder to compensate the failure in any one of turbo decoder. Parallel processing produce higher throughput compared to normal topology.

## REFERENCES

[1] Vianney Lapotre, Purushotham Murugappa, Gay Gogniat and Amer Baghdadi, A Dynamically Reconfigurable Multiprocessor Architecture for Multistandard and Multimode Turbo Decoding. *IEEE Transaction on very Large Scale Integration VLSI Systems* ***2****(4): 1-5 (2015).*

[2] Murugappa, P., R. Al-Khayat, A. Baghdadi and M. Jézéquel, A flexible high throughput multi-ASIP architecture for LDPC and turbo decoding. *Proc. Design, Autom., Test Eur. Conf. Exhibit. (DATE)*, Pp. 1–6 (2011).

[3[ Lin, C.H., C.Y.Chen, E.J.Chang and A.Y. Wu, Reconfigurable parallel turbo decoder design for multiple high-mobility 4G systems. *J. Signal Process. Syst.* **73**(2): 109–122 (2013).

[4] Sun, Y. and J. R. Cavallaro, A flexible LDPC/turbo decoder architecture. *J. Signal Process. Syst.* **64** (1): 1–16 (2011).

[5] Wong, C.C. and H.C. Chang, Reconfigurable turbo decoder with parallel architecture for 3GPP LTE system. *IEEE Trans. CircuitsSyst. II, Exp. Briefs*, **57**(7): 566–570 (2010).

[6] Muller, O., A. Baghdadi and M. Jézéquel, Parallelism efficiency in convolutional turbo decoding. *EURASIP J. Adv. Signal Process. P*p. 1-11 Article ID 927920 (2010).

[7] Zhang, J. and M.P.C. Fossorier, Shuffled iterative decoding, *IEEE Trans. Commun.* **53**(2): 209– 213 (2005).

[8] Kunze,S., E. Matus, G. Fettweis and T. Kobori, A multi-user approach towards a channel decoder for convolutional, turbo and LDPC codes, *Proc. IEEE Workshop Signal Process. Syst. (SIPS) P*p. 386–391, Oct. (2010).

[9] Condo, C., M. Martina and G. Masera, VLSI implementation of a multimode turbo/LDPC decoder architecture. *IEEE Trans. Circuits Syst. I, Reg. Papers* **60**(6): 1441–1454 (2013).

[10] Condo, C., M. Martina and G. Masera, A net work on chip based turbo/LDPC decoder architecture. *Proceedings Design, Autom. Test Eur.Conf. Exhibit. (DATE) P*p. 1525–1530 (2012)