

# Flexible UDec ASIP Processor with Multimode Turbo Decoder

G. Prasad Kumar<sup>1</sup>, M. Vijaya Laxmi<sup>2</sup>

<sup>1</sup>M.Tech, PG Scholar, Department of ECE, Srikalahastheeswara Institute of Technology, Srikalahasthi-India

<sup>2</sup>Associate professor, Department of ECE, Srikalahastheeswara Institute of Technology, Srikalahasthi –India

**Abstract:** Channel decoding is a key feature of a wireless communication standard. It allows reliable data transfer targeting high throughput over unreliable communication channels. However, a channel coding techniques is typically associated to a variety of parameters and configuration options (frame size, communication channel, signal to noise ratio, etc) among channel decoding techniques, Turbo codes are frequently adopted in the recent wireless standards to reach a very low bit error rate (BER). Furthermore, the high through put requirement of emerging services imposes the efficient exploitation of different parallelism levels of the underlying algorithms supports several wireless communication standards and is integrated in a scalable and flexible multiprocessor platform, namely UDec.

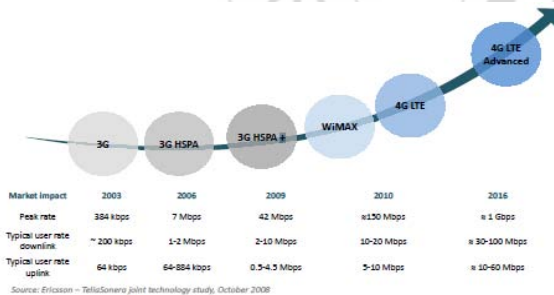
**Keywords:** Application specific instruction-set processor (ASIP), dynamic configuration, turbo codes (TCs), wireless Communication, bit error rate (BER), Universal Decoder (UDEc), Digital video Broadcasting (DVB).

## 1. Introduction

The need of mobile connectivity has hugely increased in the first decade of the 21st century. Homes, schools, businesses and people are now connected together for sharing information as soon as that information is produced. This permanent connectivity has lead to a growing number of connected mobile devices such as laptops, tablets, mobile phones, watches and plenty of other portable devices. This multiplication of connected devices goes along with a large variety of applications and traffic types needing diverse requirements.

and other mobile devices. Diverse features such as ultra broadband Internet access, IP telephony, gaming services, and streamed multimedia are provided. In order to enable such advanced services at the algorithmic level, new state of the art data processing techniques have been developed and adopted in the emerging wireless communication standards.

Channel decoding is a key feature of a wireless communication standard. It allows reliable data transfer targeting high throughput over unreliable communication Channels. However, a channel coding technique is typically associated to a variety of parameters and configuration options (frame size, communication channel, signal to noise ratio, etc). Among channel decoding techniques, Turbo codes are frequently adopted in the recent wireless standards to reach a very low bit error rate (BER). Furthermore, the high throughput requirement of emerging services imposes the efficient exploitation of different parallelism levels of the underlying algorithms. Each usage scenario corresponds to particular requirements for example in terms of throughput, latency, error rates, and/or others. Figure 2 gives an example of such usage scenario which corresponds to a mobile terminal supporting different services (High Definition Multimedia, Web Browsing, and Voice Conversation) at different channel conditions.



**Figure 1:** Communication standards evolution in mobile telephone networks

Accompanying this mobile connectivity evolution, the last years have seen considerable evolutions of wireless communication standards in the domain of mobile telephone networks, local/wide wireless area networks, and Digital Video Broadcasting (DVB). Fig1 shows the evolution of standards for mobile since 2003 in terms of throughput requirements. Besides the increasing requirements in terms of throughput and robustness against destructive channel effects, the convergence of services in single smart terminal becomes a crucial and challenging feature.

As an example, the fourth generation (4G) of cellular wireless standards aims at providing mobile broad band solution to laptop computer wireless modems, smart phones,

## 2. Problem Statement

Intensive research has been conducted to provide flexible Turbo decoder targeting high throughput, multi mode, multi standard and power consumption efficiency. However, flexible Turbo decoder implementations are not often designed regarding dynamic reconfiguration issues in the context of high throughput, multi mode and multi standard scenario requiring high speed configuration switching.

As a base architecture, we consider an ASIP based flexible Turbo decoder developed at the Electronics Department of Telecom Bretagne in Brest. The considered ASIP, namely DecASIP, supports several wireless communication

standards and is integrated in a scalable and flexible multiprocessor platform, namely UDec.

**Configuration optimization of the flexible DecASIP processor:**

- Proposal of efficient configuration parameters storage. Optimization of the configuration memory organization in order to provide a low latency configuration information transfer.
- Proposal of the support of multi configuration storage and high speed re initialization of the ASIP.
- Proposal of a generic program in order to reduce the Configuration load.

**Design of a configuration infrastructure for the UDec multi ASIP architecture:**

- Optimizations of the platform controller and the Inter connection structure of the UDec architecture in order to increase its flexibility.
- Implementation of a complete configuration Infra structure for high speed configuration of the multi ASIP UDec architecture.

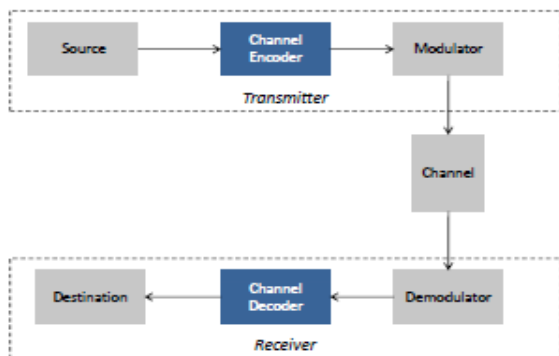
**Configuration management of the UDec architecture:**

- Definition of a configuration management where Configuration information is stored in a global configuration memory.
- Proposal of two configuration management techniques Where configuration information is generated at runtime.
- These last contributions have not been yet published. Several papers are currently under revision and will be submitted soon.

**3. Turbo Channel Decoder Design:**

**A.Context of Channel Coding:**

Channel coding techniques are used in order to reduce the noise disturbances effects by introducing redundant information to the original message. These coding techniques seek to increase as much as possible the correction capabilities of the communication system to reach the theoretical limits defined by Shannon.



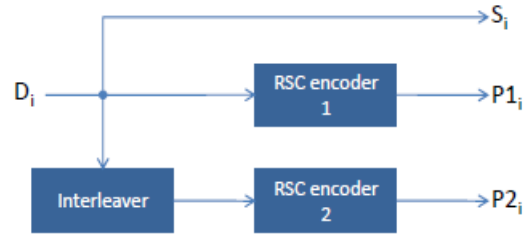
**Figure 2:** Elements of Digital communication systems

**B. Turbo Encoding**

A Turbo encoder is usually built from the parallel concatenation of two Recursive Systematic Convolutional (RSC) encoders separated by an inter leaver as shown in below figure. The first RSC encoder receives the input data

bits stream  $D_i$  in a natural order while the second RSC encoder receives the data in an interleaved one. Three output streams are generated: the systematic  $S_i$ , which is identical to the input stream and two parities  $P_{1i}$  and  $P_{2i}$  generated by the encoders in natural.

In recent standards, we observe two types of RSC encoders: the Double Binary Turbo Code (DBTC) encoder and the Single Binary Turbo Code (SBTC) encoder. The DBTC encoder generates double binary symbols by encoding bit pairs of the incoming data bits stream while the SBTC encoder encodes bitwise the incoming data bits stream.



**Figure 3:** Turbo encoder

**C.Turbo Codes Interleavers**

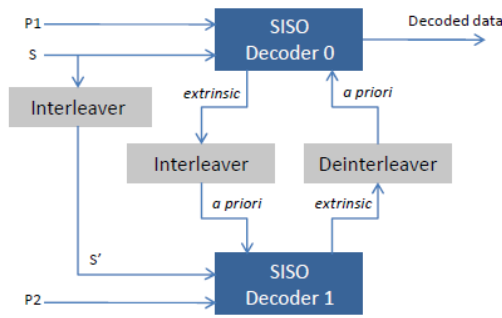
Interleavers provide an efficient solution to enhance the protection of data against destructive channel effects. For that purpose, the data is temporally dispersed. In the context of Turbo codes, the parallel concatenation of two RSC encoders provides two copies of the same symbol at different intervals of time thanks to the inter leaver that separates the two encoders. This solution allows retrieving at least one copy of the symbol if the second one has been distorted during the transmission. An inter leaver (II) satisfying this property can be verified by Studying the dispersion factor  $S$  given by the minimum distance between two symbols  $i$  and  $j$  in natural order and interleaved order.

$$S = \min_{i,j} (|i - j| + |\pi(i) - \pi(j)|)$$

The design of interleavers respecting a dispersion factor can be reasonably achieved through the S-random algorithm proposed. However, even if this kind of inter-leaver can be sufficient to validate the performance in the convergence zone of a code, it does not achieve a Good asymptotic performance. Therefore to improve the latter, the design of the inter leaver must also take into account the nature of component encoders.

**D.Turbo decoding**

Turbo decoding principle is based on an exchange of probabilistic information, called extrinsic information between two (or more) component decoders dealing with the same received set of data. As shown in Figure 4, a typical Turbo decoder consists of two decoders operating iteratively on the received frame.

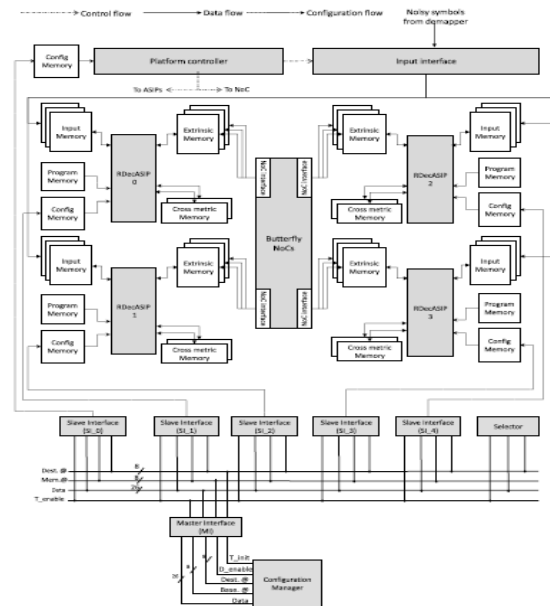


**Figure 4:** Typical turbo decoder

The first component (SISO decoder 0 in Figure 4) works in natural domain while the second (SISO decoder 1 in Figure 4) works in interleaved domain. The Soft - Input Soft-Output (SISO) decoders operate on soft information to improve the decoding performance.

#### 4. UDec Architecture

The proposed dynamic reconfigurable UDec turbo decoder architecture is shown in Fig. 6. It consists of two rows of RDecASIPs interconnected via two butterfly topology networks on chip (NoCs). Each row corresponds to a component decoder. In the example of Fig. 6, four ASIPs are organized in two component decoders, respectively, built with two ASIPs. Within each component decoder, the ASIPs are connected by two 44-bit buses for boundary state metrics exchange (not shown in Fig. 6). The RDecASIP implements the Max-Log-MAP algorithm. It supports both single and double binary convolutional TCs. Moreover, sliding window technique large frames are processed by dividing the frame into  $N$  windows, each with a maximum size of 64 symbols. Each ASIP can manage a maximum of 12 windows. Each ASIP can be configured through a  $26 \times 12$  configuration memory. Since the RDecASIP is designed to work in multi ASIP architecture as described. It requires several parameters to deal with a sub block of the data frame and several parameters to configure the ASIP mode. Concerning the sub block partitioning, each ASIP is configured with the size and the number of windows it has to decode. Furthermore, the last window size can be different, and so it corresponds to an additional parameter. In a SBTC mode, the address of the tail bits in memory, the size, and the number of windows for the tail bits have to be configured. Parameters for the ASIP mode correspond to the location of the ASIP in the architecture, the number of ASIPs required, the parameter that defines if the current ASIP is in charge of tail bits or not.



**Figure 5:** Reconfigurable UDec system architecture example with four ASIPs

The entire configuration load represents 253 bits. Moreover, RDecASIP is associated with three memory banks of size  $24 \times 256$  used to store the input channel Log-Likelihood Ratio values. There are also three banks of size  $30 \times 256$  used for extrinsic information storing. Each ASIP is further equipped with two  $88 \times 32$  memories, which hold 11-bit state metric values and a  $16 \times 128$  program memory that stores a unified program for turbo decoding modes. In a pipeline unidirectional bus implementing incremental burst, multicast, and broadcast mechanisms. It can be split in three functional blocks: 1) master interface (MI); 2) slave interface (SI); and 3) selector. Each configuration memory is connected to the bus through an SI. The configuration manager deals with the configuration generation that is based on internal decisions and external information and commands. The MI provides an interface allowing the connection of the configuration manager to the bus. The SI provides an interface between the bus and the configuration memory. The selector provides a simple and efficient solution to select, at run-time, RDecASIPs that are targeted by the next configuration data. This solution allows the transfer of a data into the configuration memory with a latency of 5 clock cycles.

#### 5. Flexible UDec Architecture

This section presents several techniques that we propose in order to increase the dynamic configuration ability in the context of multi ASIP platform for flexible turbo decoding. These techniques concern the communication networks connecting the ASIPs and the multi ASIP platform controller.



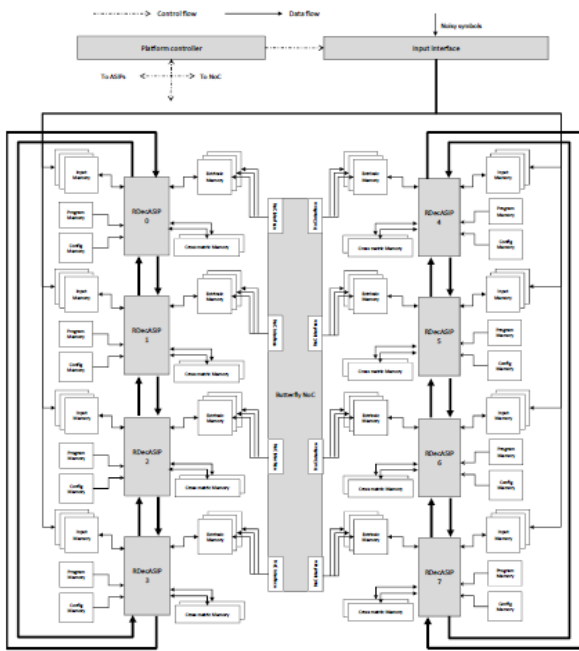


Figure 6: Flexible UDec architecture

This specific number of RDecASIPs has been chosen in order to be able to illustrate all dynamic configuration issues that the UDec architecture has to face while keeping reasonable complexity for the clarity of the presentation. In fact, the contributions presented in this chapter remains valid for lower and higher number of RDecASIPs. In the following section, the possibility to adapt at run time the number and the location of the active RDecASIPs used for a given configuration is addressed.

**a. ASIP number and location:**

In a multi mode and multi standard context, the requirements in terms of throughput and BER evolve dynamically. Thus, depending of these requirements, the number of activated ASIPs has to be adapted at runtime. Moreover, in order to deal with hot spot and potentially faulty cores management for the UDec architecture, the location of the activated ASIPs has to be dynamically defined. Obviously, this new flexibility impacts the different components of the architecture. In the initial UDec architecture, the number of ASIPs used for a given Configuration was fixed at design time and was equal to the total number of implemented Cores. Therefore, the two ring buses and the Butterfly topology NoCs did not support a dynamic evolution of the number and location of the ASIPs selected for a given configuration.

**b. Ring buses adaptation:**

The ring buses consist of direct connections between the ASIPs allowing exchanging boundary state metrics as shown in Figure 8. So, when the number and the location of the selected ASIPs dynamically evolve, the loop connections between the last and the first selected ASIPs have to be adapted. Figure 9 shows different examples of the ring buses adaptation for one component decoder. Figure 9(a) shows the case where only one ASIP is used in the component decoder while Figure 9(b) shows the case where two ASIPs are selected to perform the decoding task. Moreover, the location of the first ASIP has been shifted from RDecASIP 0 to

RDecASIP 1. Finally, Figure 9(c) shows the case where three ASIPs are selected and the location of the first ASIP has been shifted from RDecASIP 0 to RDecASIP 2.

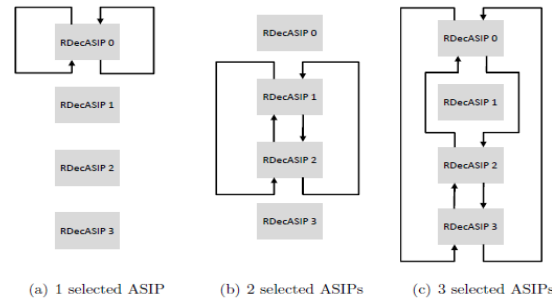


Figure 7: Ring Buses Adaption

In this case, the last ASIP of the component decoder becomes the RDecASIP 0. However, in order to reduce the design and the management complexity of this new flexibility feature, the selected ASIPs have to be physically consecutive in the component decoder as shown in the examples.

**6. Results**

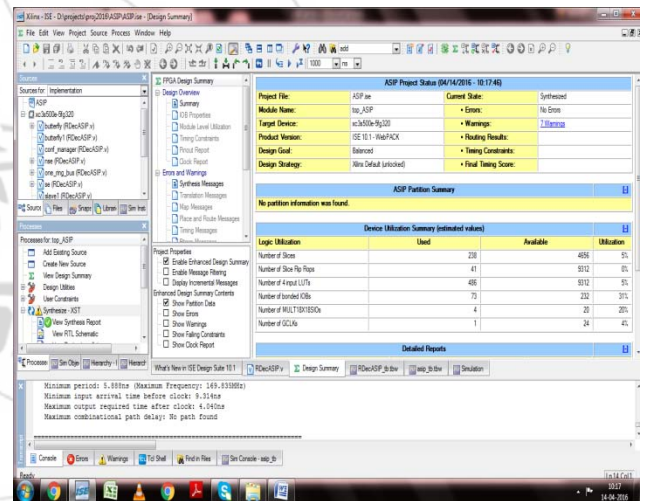


Figure 8 : Design summary of flexible UDEC ASIP's

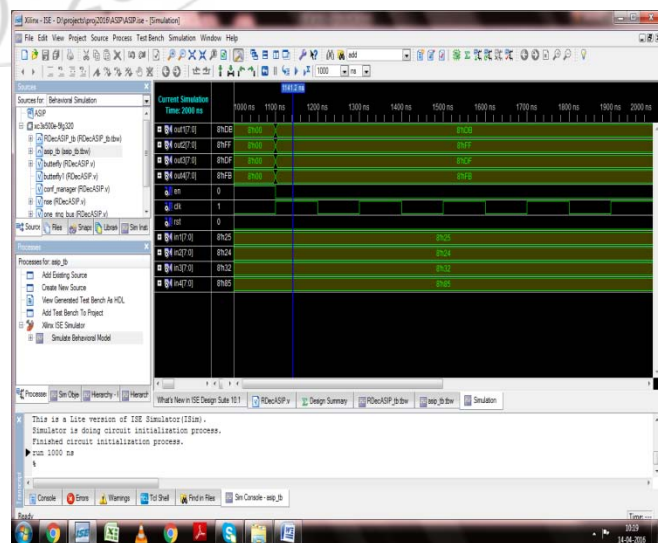


Figure 9: Simulation Results of Flexible UDEC ASIP's

No.of processors	Power (mW)	Delay (ns)
4-ASIP's	82	5.888
8-ASIP's	82	4.040

**Figure 10:** power and delay results

## 7. Conclusion

Channel coding is a key feature of a wireless standard allowing reliable data transfer. Among channel coding techniques, Turbo codes are frequently adopted to reach a very low bit error rate. Moreover, the multiplication of communication standards leads to complex scenarios where the configuration process becomes a key point in order to guarantee high performances. In fact, most of the existing related works have proposed flexible hardware platforms while trying to optimize their efficiency in terms of area, throughput, and energy consumption. These optimizations lead to a configuration load reduced by 70% compared to the initial ASIP. Results show that a dedicated memory organization taking into account the multiprocessor context hugely reduces the configuration load (i.e. more than 90%) when a configuration infrastructure implementing multicast and broadcast mechanisms is used. Logic synthesis results targeting 65 nm CMOS technology show that these optimizations introduce a low area overhead of 0.009 mm<sup>2</sup> while the decoding performance and maximum clock frequency of the RDecASIP have remained identical to the initial implementation.

## References

- [1] C.E.Shannon, \A mathematical theory of Communication ,\" Bell system technical journal, vol. vol. 27, 1948.
- [2] \3GPP TS 36.212: Multiplexing and channel coding, version 8.4.0,\" Sept. 2008.
- [3] C. Berrou, A. Glavieux, and P. Thitimajshima, \Near Shannon limit error correcting coding and decoding: Turbo-codes. 1,\" in IEEE International Conference on Communications, ICC 93., vol. 2, 1993, pp. 1064{1070 vol.2.
- [4] IEEE Standard for Local and Metropolitan Area Networks Part 16: Air In-terface for Fixed and Mobile Broadband Wireless Access Systems, Std., 2006.
- [5] S. Lin and D. Costello, Jr., Error Control Coding: Fundamentals and Applications. Prentice Hall, Englewood Cliffs, NJ., 1983.
- [6] C. Douillard, M. Jezequel, C. Berrou, J. Tusch, N. Pham, and N. Brengarth, \The Turbo Code Standard for DVB-RCS,\" in 2nd International Symposium on Turbo Codes & Related Topics, Brest, France, 2000, pp. 535 538.
- [7] S. Dolinar and D. Divsalar, \Weight distributions for turbo codes using random and nonrandom permutations,\" in The Telecommunications and Data acquisition Report, Tech. Rep., 1995, pp. 56{65.
- [8] J. Hagenauer and P. Hoehner, \A viterbi algorithm with soft-decision outputs and its applications,\" in IEEE Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond' (GLOBECOM), 1989, pp. 1680{1686 vol.3.

## Author Profile



**Prasad kumar G** received B. Tech degree in electronics and communication engineering from JNTUA University, Andhra Pradesh & pursuing M.Tech in Electronics and Communication Engineering in JNTUA, Andhra Pradesh.

**Vijaya Laxmi M**, Associate Professor, Electronics and Communication Engineering, Srikalahastheeswara institute of technology, Andhra Pradesh