ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 9, April 2017

Reconfigurable Multi-ASIP Architecture for Turbo Decoding

KOTTE USHA SRI [1]
kotteushasree25@gmail.com[1]

M.KIRAN[2]
kiran_m_20@rediffmail.com

[1]PG Scholar, Dept of ECE, Gokaraju Rangaraju Institute of Engineering and Technology, Nizampet, Bachupally, Kukatpally ,Hyderabad ,Telangana.

[2]Associate Professor, Dept of ECE, Gokaraju Rangaraju Institute of Engineering and Technology, Nizampet, Bachupally, Kukatpally , Hyderabad ,Telangana.

Abstract- **The multiplication of wireless communication standards in the domain of mobile telephone networks, local/wide wireless area networks, and Digital Video Broadcasting (DVB), is introducing the need of flexible and reconfigurable multistandard baseband receivers. Application Specific Instruction-set Processor (ASIP) has a general-purpose architecture that can be modified and used in a variety of applications. However, this increases the power and memory usage that affects the functionality and efficiency of ASIP. The development of turbo codes has allowed Shannon limit information transfer, in modern communication systems. In this context, multiprocessor turbo decoders are developed in order to support the increasing flexibility and throughput requirements of arising applications. However, these solutions are inefficient to address reconfiguration performance issues, which can be a limiting factor in the forecoming years . This paper presents the design of a reconfigurable multiprocessor architecture for turbo decoding achieving very fast reconfiguration without conceding the decoding performances.**

***Index Terms*— Application specific instruction-set processor (ASIP), dynamic configuration, turbo codes (TCs), wireless communication.**

I. INTROUCTION

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). 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. Recently, there are a large growing emergence and demand for an inexpensive and ubiquitous broadband wireless network. Thus, Long Term Evolution (LTE) and WiMAX standards become prevalent for the broadband wireless network. Meanwhile, fourth generation (4G) cellular wireless communication, the term referred to International Mobile Telecommunications-Advanced (IMT-Advanced), is emerging in high-end broadband wireless devices. Nowadays the 4G compliant versions of LTE and WiMAX are

LTE Advanced (LTE-A) and WirelessMAN-Advanced (WiMAX-2), respectively. To achieve a smooth migration for different applications, a CTC decoder that works across the dual IMT-Advanced compliant standards is necessary. Hence, the goal of this work is to design a CTC decoder that can be used in high-mobility 4G communications.

The introduction of contention-free interleavers in recent communication standards, such as WiMAX and LTE, enables high-throughput implementations These architectures propose to use multiple Soft-Input Soft-Output (SISO) decoders to reach the high-throughput requirement of emerging and future standards. These turbo decoders offer certain degrees of flexibility to adapt for instance the number of SISO decoders, the TC mode, i.e., single binary TC (SBTC) or double binary TC (DBTC), or the FS. However, these efforts do not present any configuration infrastructures associated to these architectures in order to support fast and efficient dynamic configuration switches. Recently, application specific instruction set processor (ASIP) solutions have been investigated in order to offer architectures providing good tradeoffs in terms of flexibility, throughput, and power dissipation.

The rest of this brief is organized as follows. Section II gives more insights about the motivation of this brief. Section III introduces the considered multi-ASIP architecture implementing the RDec ASIP processor. Section IV presents the flexibility features added to the initial architecture.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 9, April 2017

Section V presents the implementation results. Finally, Section VI concludes this brief.

II. MOTIVATION

When a turbo decoder is designed to support several communication standards, the decoder behavior has to be dynamically adapted in order to respect the application requirements and to take into account the communication channel quality. In this paper, the scenario example presented in Fig. 1 is considered as the worst-case configuration scenario that should be met by a multi-mode and multi-standard turbo decoder in mobility.

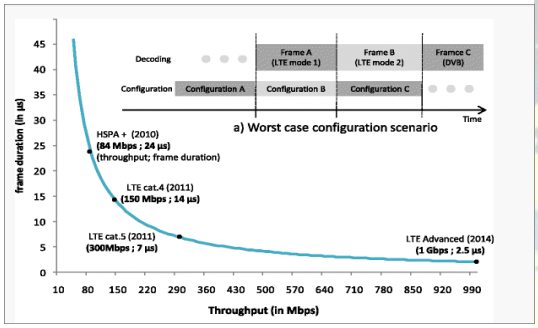

Fig. 1 Decoding latency of a 2048-bit frame.

A. *Reviews of Turbo Codes*

Background reviews of the CTCs of WiMAX-based standard and LTE-based standard are demonstrated in this section. Turbo Codec The encoder consists of two identical recursive systematic convolutional (RSC) encoders, which are connected in parallel by an interleaver. The RSC encoder produces systematic symbol (us) and parity symbol (up) to the channel. The information stream is reordered by the interleaver and then enters the second RSC encoder. The RSC encoders of SBCTC encodes 1 bit per time while the RSC encoders of DBCTC encodes 2 bits per time. The RSC encoders of LTE (SB-CTC) and WiMAX (DB-CTC).For the trellis termination, the RSC encoder in LTE uses a few redundant bits to terminate the trellis path. However, the RSC encoder in WiMAX adopts two-phase circular encoding [3] where the final trellis state can be the same to initial trellis state. The decoder is composed of two soft-input soft-output (SISO) decoders that are serially concatenated by the interleaver and de-interleaver. Each SISO decoder uses the received systematic symbol and corresponding parity symbol to computes extrinsic information that is then iteratively fed to the other SISO decoder as the a priori information. The hard decision of decoded symbols is made after several iterations between these two SISO decoders. The interleaver is one of dominant modules of CTCs and the correction performance of CTCs depends on the structures and length of interleaver. The interleaver is used to permute the order of symbols. This can be done by using an interleaving address generator to access the symbols stored in a buffer.

III. UDEC ARCHITECTURE

The proposed dynamically reconfigurable UDec architecture considered in this work is shown in Fig. 2. It consists of two columns of RDec ASIP processors interconnected via two unidirectional Butterfly networks-on-chip supporting shuffled turbo decoding. Each column corresponds to a component decoder, one processing the data in natural domain and the other in interleaved domain. Each RDecASIP is associated with three input memory banks of size 24 × 256 used to store the input channel LLR values.

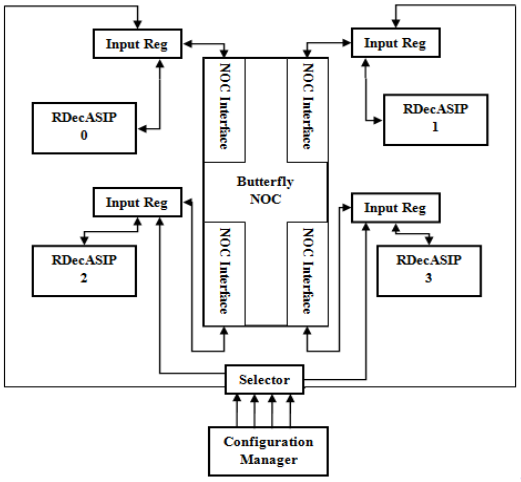

Fig. 2 Reconfigurable UDec system architecture example with 2 × 2 ASIPs

The platform is dynamically configured through a dedicated bus-based communication infrastructure shown in Fig. 2 that consists in a pipeline unidirectional bus implementing incremental burst, multicast, and broadcast mechanisms. It can be split in three functional blocks:

1) master interface (MI);

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 9, April 2017

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. Moreover, because of the pipeline nature of the transfers, the configuration infrastructure is able to provide one data per clock cycle to the destination. The rest of this brief focuses on the reconfiguration process of the proposed multi-ASIP architecture through methods, which bring more flexibility and more efficiency to the UDec platform.

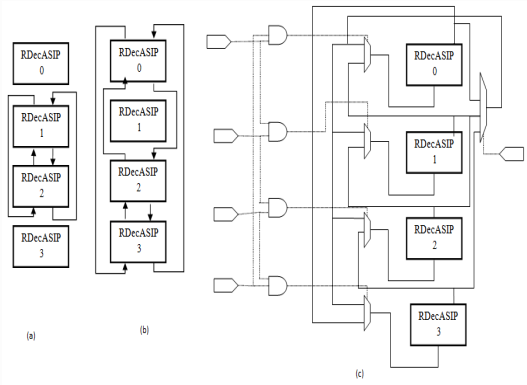

Fig. 3. Ring buses dynamic adaptation examples and architecture. (a) Two selected ASIPs. (b) Three selected ASIPs. (c) Flexible architecture illustrated for one ring bus.

IV. FLEXIBLE UDE ARCHITECTURE

This section presents the techniques that we propose in order to increase the dynamic configuration ability of the UDec architecture. In a multimode and multistandard context, the requirements in terms of throughput and BER evolve dynamically. Thus, depending on these requirements, the number of active ASIPs has to be adapted at run-time. Moreover, in order to deal with hotspot and potentially faulty cores management for the UDec architecture, the location of the activated ASIPs has to be dynamically defined. In the initial UDec architecture, the number of ASIPs used for a given configuration is fixed at design time and is equal to the total number of implemented cores. Sections IV-A and IV-B present solutions in order to bring this new flexibility.

A. *Ring Buses Adaptation*

The ring buses consist of direct connections between the ASIPs allowing exchanging boundary state metrics. 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. Fig. 2 shows different examples of the ring buses adaptation when four ASIPs are implemented in each component decoder. Fig. 2(a) shows the case where two ASIPs are selected to perform the decoding task. The location of the first ASIP has been shifted from RDecASIP 0 to RDecASIP 1. Fig. 2(b) shows the case where three ASIPs are selected and the location of the first ASIP has been shifted from RDecASIP 0 to RDecASIP 2. In this case, the last ASIP of the component decoder is the RDecASIP 0, and the RDecASIP 1 has to be bypassed.

B. *Butterfly Topology NoCs Adaptation*

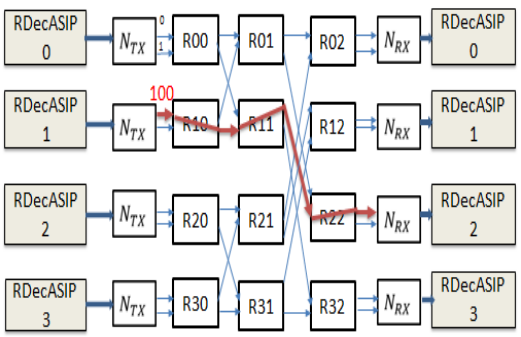

Fig. 4. Butterfly topology routing principle.

The extrinsic information transfers through the NoC are also impacted when the location of the selected ASIPs changes dynamically. Indeed, the routing information for the transfer is computed by the network interface associated with each ASIP depending on a global address of the symbol generated by the ASIP. Fig. 4 illustrates the routing principle for the considered butterfly topology NoC. This topology allows a unique path in the network between each pair of nodes (source to destination). For each router, there are two inputs and two outputs. A single bit is used in a router to select the appropriate output:

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 9, April 2017

1) 0 for the first output and
2) 1 for the second output.

In the proposed architecture, the routing information has to be adapted depending on the location of the ASIPs determined by the ASIP Shift value and the level of sub block parallelism determined by the number of selected RDecASIPs for the configuration in each component decoder (NumASIPs).

Convolutional codes

Convolutional codes offer an approach to error control coding substantially different from that of block codes.

A convolutional encoder:

- encodes the entire data stream, into a single codeword.
- does not need to segment the data stream into blocks of fixed size (Convolutional codes are often forced to block structure by periodic truncation).
- is a machine with memory.
- This fundamental difference in approach imparts a different nature to the design and evaluation of the code.
- Block codes are based on algebraic/combinatorial techniques.
- Convolutional codes are based on construction techniques.
- A Convolutional code is specified by three parametersor where is the coding rate, determining the number of data bits per coded bit.
- In practice, usually k=1 is chosen and we assume that from now on.K is the constraint length of the encoder a where the encoder has K-1 memory elements.
- There is different definitions in literatures for constraint length

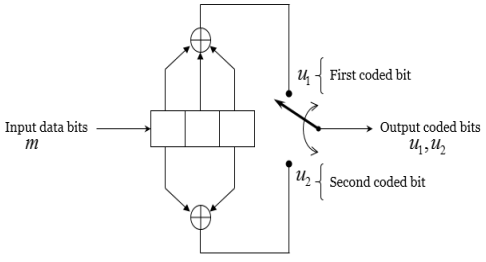

Fig 5 A Rate ½ Convolutional encode

Convolutional encoder (rate ½, K=3) 3 shift-registers where the first one takes the incoming data bit and the rest, form the memory of the encoder.

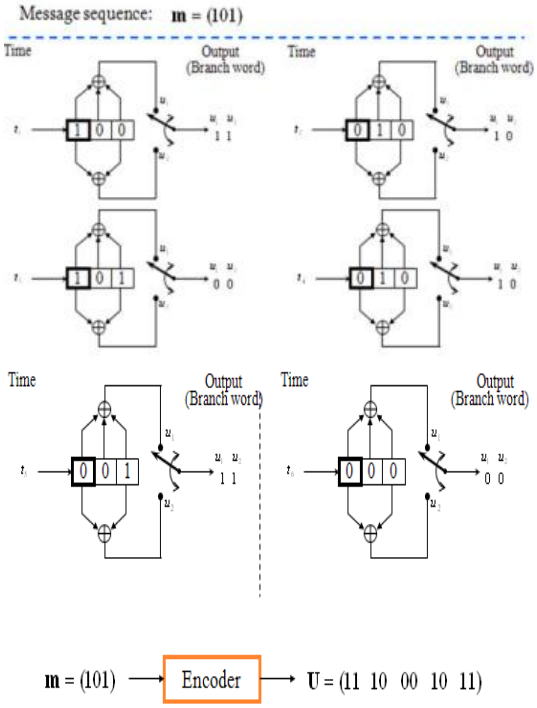

Fig.6 Convolutional encoder (rate ½, K=3)

The coding trellis for each is given by the figure. The blue lines show transitions in response to a 0 and red lines in response to a 1. The notation 1/11, the first number is the input bit, the next are two code bits. Of these, the first is what we called the systematic bit, and as you can see, it is same as the input bit. The second bit is the parity bit. Each code uses this trellis for encoding. [5] proposed a system, Low Voltage Differential Signaling (LVDS) is a way to communicate data using a very low voltage swing (about 350mV) differentially over two PCB traces. It deals about the analysis and design of a low power, low noise and high speed comparator for a high performance Low Voltage Differential Signaling (LVDS) Receiver. The circuit of a Conventional Double Tail Latch Type Comparator is modified for the purpose of low-power and low noise operation even in small supply voltages. The circuit is simulated with 2V DC supply voltage, 350mV 500MHz sinusoidal input and 1GHz clock frequency. LVDS Receiver using comparator as its second stage is designed and simulated in Cadence Virtuoso Analog Design Environment using GPDK 180nm .By this design,

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 9, April 2017

the power dissipation, delay and noise can be reduced.

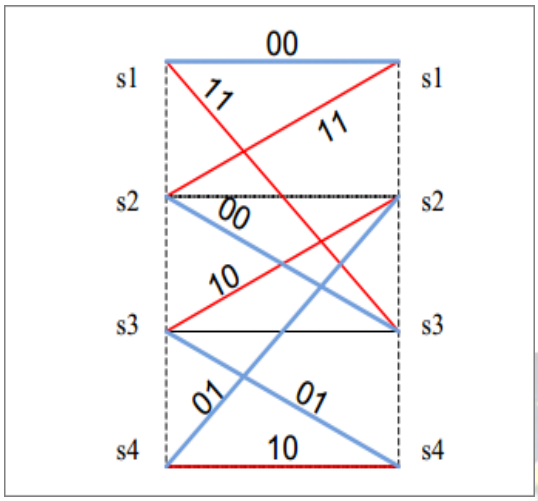

Fig.6 The trellis diagram

V. SIMULATION RESULTS

The simulation result of the 256-bit Reconfigurable multi-ASIP is shown in Fig.7. The total inputs to the 256-bit Reconfigurable multi-ASIP are 256 and we get 256-bit output. The Reconfigurable multi-ASIP is synthesized under the constraint of 600MHz in Xilinx ISE 13.2 targeting for Spartan3E XC3S500E device FT256 package with a speed grade of -5. Power is measured by Xilinx XPower analyzer using the switching activity interchange format file recorded in a sufficient long simulation time.

256-bit Reconfigurable multi-ASIP

A. Simulation

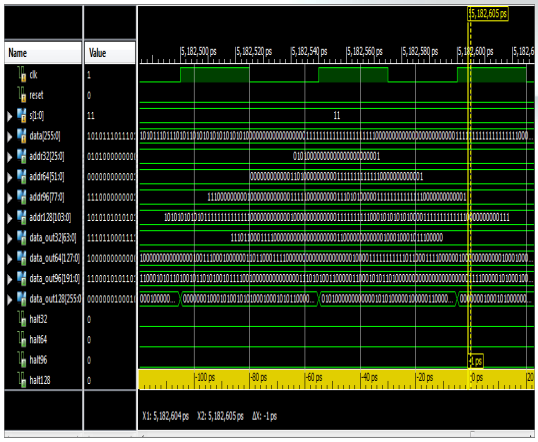

Fig.7 Simulation Result of 256-bit Reconfigurable Multi-ASIP

B. Design Summary

TABLE I
DESIGN SUMMARY OF 256-BIT RECONFIGURABLE MULTI-ASIP

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	4039	4656	86%
Number of slice flip Flops	1948	9312	20%
Number of 4 input LUTs	7026	9312	75%
Number of bonded IOBs	117	190	62%
Number of GCLKs	1	24	4%

C. RTL Schematic

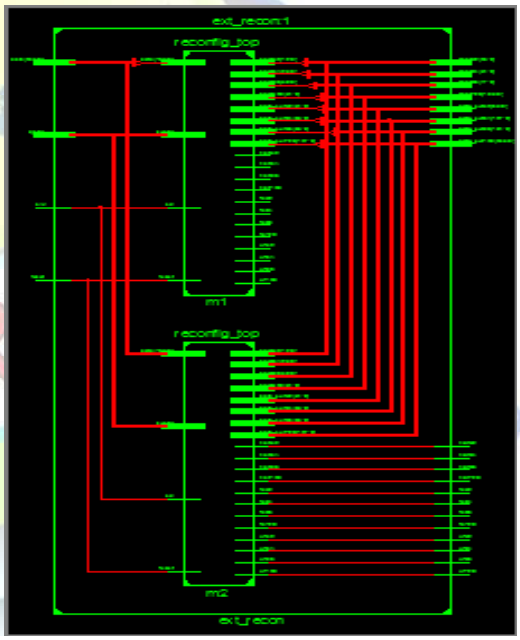

Fig.8 Synthesis Result of 256-bit Reconfigurable Multi-ASIP

128-bit Reconfigurable multi-ASIP

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 9, April 2017

A. Simulation

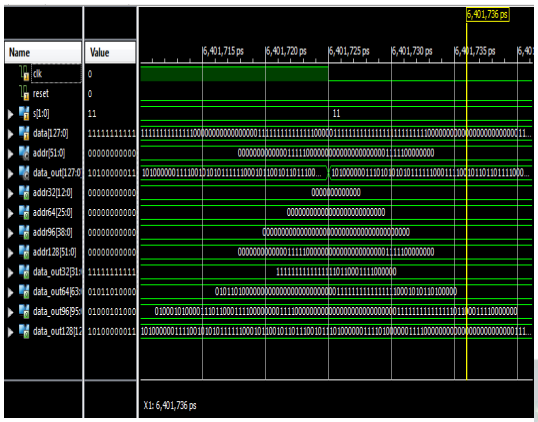

Fig.9 Simulation Result of 128-bit Reconfigurable Multi-ASIP

B. Design Summary

TABLE II
DESIGN SUMMARY OF 128-BIT RECONFIGURABLE MULTI-ASIP

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	925	4656	19%
Number of slice flip Flops	918	9312	9%
Number of 4 input LUTs	1577	9312	16%
Number of bonded IOBs	59	190	31%
Number of GCLKs	1	24	4%

C. RTL SCHEMATIC

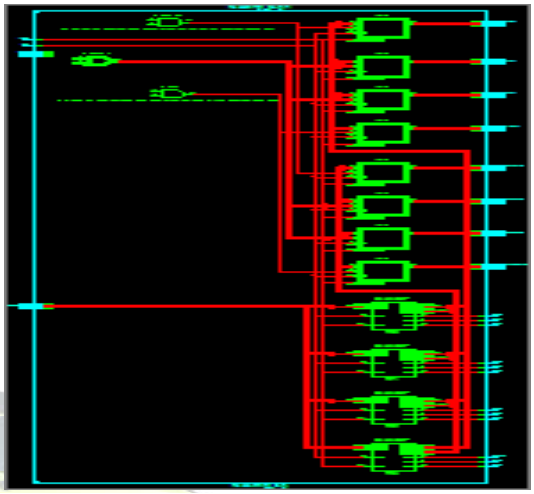

Fig.10 Synthesis Result of 128-bit Reconfigurable Multi-ASIP

Comparison results of 128-bit and 256-bit Reconfigurable multi-ASIP

TABLE III
COMPARISON RESULTS OF 256-BIT AND 128-BIT RECONFIGURABLE MULTI-ASIP'S

Parameter	128-bit Reconfigurable Multi-ASIP	256-bit Reconfigurable Multi-ASIP
Minimum Period	13.143ns	14.757ns
Minimum Input arrival time before Clock	12.327ns	15.353ns
Maximum Output required time after Clock	4.040ns	4.040ns
Memory	315.5MB	450.7MB

VI. CONCLUSION

The multiplication of communication standards leads to complex scenarios where the configuration process becomes a key point in order to guarantee high performances. In this context, this brief tackles the dynamic configuration issues of multiprocessor platform for turbo decoding in order to respect hard constraints imposed by emerging multimode and multistandard scenario Moreover, ASIC synthesis results considering up to 128 RDecASIPs have demonstrated that a configuration transfer latency below 1 μs is reachable providing an efficient solution in order to support dynamic configuration in the multimode and multistandard

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 9, April 2017

scenario. The implementation of the Turbo Decoding using Trellis Routing with 256 bit is presented in the paper.

REFERENCES

[1] O. Muller, A. Baghdadi, and M. Jézéquel, "Parallelism efficiency in convolutional turbo decoding," *EURASIP J. Adv. Signal Process.*, vol. 2010, p. 927920, Dec. 2010.
[2] J. Zhang and M. P. C. Fossorier, "Shuffled iterative decoding," *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 209–213, Feb. 2005.
[3] C.-C. Wong and H.-C. Chang, "Reconfigurable turbo decoder with parallel architecture for 3GPP LTE system," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 7, pp. 566–570, Jul. 2010.
[4] C.-H. Lin, 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.*, vol. 73, no. 2, pp. 109–122, Nov. 2013.
[5] Christo Ananth, Bincy P Chacko, "Analysis and Design of Low Voltage Low Noise LVDS Receiver", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 9, Issue 2, Ver. V (Mar - Apr. 2014), PP 10-18
[6] C.-H. Lin, C.-Y. Chen, and A.-Y. Wu, "Area-efficient scalable MAP processor design for high-throughput multistandard convolutional turbo decoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 2, pp. 305–318, Feb. 2011.
[7] G. Wang, H. Shen, Y. Sun, J. R. Cavallaro, A. Vosoughi, and Y. Guo, "Parallel interleaver design for a high throughput HSPA +/LTE multistandard turbo decoder," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 5, pp. 1376–1389, May 2014.
[8] M. May, T. Ilnseher, N. Wehn, and W. Raab, "A 150 Mbit/s 3GPP LTE turbo code decoder," in *Proc. Design, Autom., Test Eur. Conf. Exhibit. (DATE)*, 2010, pp. 1420–1425.
[9] O. Muller, A. Baghdadi, and M. Jézéquel, "ASIP-based multiprocessor SoC design for simple and double binary turbo decoding," in *Proc. Design, Autom., Test Eur. Conf. Exhibit. (DATE)*, vol. 1. Mar. 2006, pp. 1–6.
[10] P. Murugappa, R. Al-Khayat, A. Baghdadi, and M. Jézéquel, "A flexible high throughput multi-ASIP architecture for LDPC and turbo decoding," in *Proc. Design, Autom., Test Eur. Conf. Exhibit. (DATE)*, 2011, pp. 1–6.
[11] C. Brehm, T. Ilnseher, and N. Wehn, "A scalable multi-ASIP architecture for standard compliant trellis decoding," in *Proc. Int. SoC Design Conf. (ISOCC)*, 2011, pp. 349–352.
[12] S. Kunze, E. Matus, G. Fettweis, and T. Kobori, "A 'multi-user' approach towards a channel decoder for convolutional, turbo and LDPC codes," in *Proc. IEEE Workshop Signal Process. Syst. (SIPS)*, Oct. 2010, pp. 386–391.
[13] C. Condo, M. Martina, and G. Masera, "VLSI implementation of a multimode turbo/LDPC decoder architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 6, pp. 1441–1454, Jun. 2013.
[14] C. Condo, M. Martina, and G. Masera, "A network-on-chip-based turbo/LDPC decoder architecture," in *Proc. Design, Autom., Test Eur. Conf. Exhibit. (DATE)*, 2012, pp. 1525–1530.
[15] V. Lapotre *et al.*, "Optimizations for an efficient reconfiguration of an ASIP-based turbo decoder," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 493–496.
[16] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 2. Jun. 1995, pp. 1009–1013.
[17] 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.
[18] N. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-output decoding algorithms in iterative decoding of turbo codes," JPL, Pasadena, CA, USA, TDA Prog. Rep. 42-124, 1996.
[19] V. Lapotre, P. Murugappa, G. Gogniat, A. Baghdadi, M. Hubner, and J.-P. Diguet, "Stopping-free dynamic configuration of a multi-ASIP turbo decoder," in *Proc. Euromicro Conf. Digit. Syst. Design (DSD)*, 2013, pp.