Towards a reconfigurable hardware architecture for implementing a LDPC module suitable for software radio systems

Rene Cumplido[1], Juan Campos[2], Claudia Feregrino-Uribe[3], Roberto Perez[4]

[1,2,3] National Institute for Astrophysics Optics and Electronics, Computer Science
Department, Luis Enrique Erro 1, Sta. Ma. Tonantzintla, 72840, Puebla, Mexico.
[4] Center for Scientific Research and Higher Education of Tamaulipas
Scientific and technologic park TECNOTAM, Km. 5.5 Victoria-Soto La Marina 87130
Victoria city, Tampaulipas, Mexico.
([1]rcumplido, [2]jcampos, [3]cferegrino)@inaoep.mx,
[4]jrperez@tamps.cinvestav.mx

Abstract. Forward Error Correction is a key piece in modern digital communications. When a signal is transmitted over a noisy channel, multiple errors are generated. FEC techniques are directed towards the recovery of such errors. In last years, LDPC (Low Density Parity Check) codes have attracted attention of researchers because of their excellent error correction capabilities, but for actual radios high performance is not enough since they require to communicate with other multiple radios too. In general, communication between multiple radios requires the use of different standards. In this sense, Software Defined Radio (SDR) approach allows building multi standard radios based on reconfigurability abilities which means that base components including recovery errors block must provide reconfigurable options. In this paper, some open problems in designing and implementing reconfigurable LDPC components are presented and discussed. Some features of works in the state of the art are commented and possible research lines proposed.

Keywords. LDPC codes, SDR, Software Defined Radio, Hardware Implementation

1 Introduction

The fast development of communication technologies has made possible the availability of multiple devices that offer a wide range of services based on novel technologies. However, the development of new schemes of communication creates new challenges for providing smaller, cheaper and faster devices. According to market requirements and with the emergence of new standards and protocols, wireless systems manufacturers and service providers must respond to changes as they occur by upgrading systems to incorporate the latest innovations or to fix bugs as they are discovered. Since frequent redesign is expensive, time-consuming and inconvenient to end users, future-proof radios are an interesting

Dagstuhl Seminar Proceedings 10281
Dynamically Reconfigurable Architectures
http://drops.dagstuhl.de/opus/volltexte/2010/2895

option. Such radios can be efficiently implemented using software radio architectures in which the radio reconfigures itself based on functionalities it will be supporting [1]. Forward Error Correction (FEC) is a key piece in any modern communication system since it allows to modern communication systems to work very close to theoretical Shannon limit [2], in other words, using FEC techniques it is possible to transmit digital data with high reliability over noisely corrupted channels by encoding the digital message prior the transmission. Finally, encoded information is used at the receiver for recovering the original information.

In the family of techniques used in error correction, LDPC (Low Density Parity Check) codes have focused the interest of the coding community since it presents a BER (Bit Error Rate) very close to the Shannon limit. The easiest way for building radios supporting multiple standards is to build a specific architecture for each standard and just select one of them, obviously this solution requires a high implementation area and is no optimum in key aspects as power consumption and use of memory resources which are crucial aspects in mobile devices. Due this issues, there exists an interest in reconfigurable architectures able of reuse components or exploit similar aspects in algorithms for implementing different Error Correction codes using the minimum possible area, specifically for LDPC. For instance, [7] presents a multi standard design including Viterbi, turbo codes an LDPC decoders based on memory share, [8] proposes an LDPC decoder for DVB-S2 with reconfiguration capabilities only in design time, [3] presents an LDPC decoder architecture that supports variable block sizes and multiple code rates based on a barrel shift communication network and [4] proposes the use of multiple communication buses focusing in latency caused by the communication between components.

2 LDPC codes

In general there are two different approaches for representing an LDPC code:
1.- Using a matrix (as in all block codes) called parity check matrix (H) which represents the restrictions set. Parity check matrix (H) contains only a very small number of non-zero entries. The sparseness of H is essential for achieving low complexity decoding process. In general, decoding process increases only linearly respect to code length.
2.- Using Tanner graphs. These graphs not only provide a complete representation of the code, they also help to understand the decoding algorithm as explained later in this section. Tanner graphs are bipartite graphs with nodes separated into two different sets. These sets are connected using edges and represent code restrictions. The two types of nodes in a Tanner graph are called: Variable nodes (v-nodes), every v-node corresponds to every bit in the code code word and check nodes (c-nodes), every c-node corresponds to every parity check equation used for encoding the original data word. Possible representations for an (8,4) code are shown in Fig. 1.

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

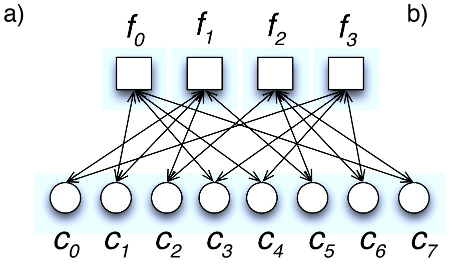

Fig. 1. Representation of LDPC codes. a) Parity-check matrix b) Tanner graph

The class of algorithms used to decode LDPC codes are collectively called message-passing algorithms since their operation is based on the interchange of messages between nodes in the representative Tanner graph. Messages pass back and forward between v-nodes and c-nodes iteratively until a result is achieved or the process is halted. Specifically, these algorithms are named according to the type of passed message or to the type of operation performed at the nodes. For instance, belief-propagation where the messages between nodes are probabilities (or log likelihood ratios) and sum-product where the performed operations at the c-nodes are basically sums and products.

3 Hardware Implementation

Traditionally, LDPC decoders have been implemented using fully parallel and semi parallel approaches. Using a fully parallel approach, every node in the Tanner graph is processed in parallel fashion, thats means that all operations in nodes are executed at the same clock time. In general, a fixed communication net between nodes is designed and no reconfigurability options are allowed. As expected, these implementations does present high throughput rates but large implementation size and high power consumption. The majority of parallel implementations use custom architectures due to the high complexity required for the interconnection net between nodes.

A semi parallel implementation is a better and more realistic trade off between use of area resources and throughput, in this case just a fixed number of nodes are implemented and reused for processing the complete Tanner graph. In these implementations, the number of memory accesses is directly related to the number of nodes that are processed in parallel because messages between v-nodes and c-nodes must be temporary stored. Since the amount of information to be stored and accessed is high, memory collisions could be a significant issue if the routing net it is not correctly designed.

When the requirement is a reconfigurable architecture for LDPC decoding, all the last considerations are increased in complexity and it is necessary to look for new approaches in order to facilitate the design and the implementation of such required system. Reconfiguration process requires a full reconnection between nodes according to the selected H matrix, it presents a high complexity degree specially when capacities must provide support to non-structured LDPC codes. Depending on the nodes design, they could require some changes when reconfiguration is activated caused by the variable number of connected edges. This depends on the selected H matrix.

Independently of the selected approach for implementing an LDPC decoder, implementation of three basic processes must be faced :

1. Operations in nodes.
2. Memory data access.
3. Routing net between nodes.

Every process provides a specific functionality necessary for the correct operation of the entire system. A basic overview of the general decoder architecture is shown in Fig. 2.

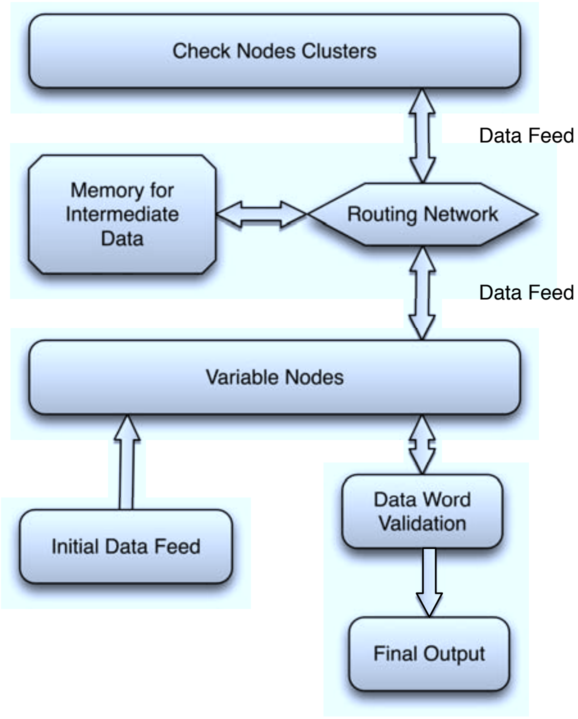

Fig. 2. High level of a Reconfigurable Architecture for LDPC decoding

3.1 Operations in nodes

C-nodes require high computational power since they are responsible of calculations for correlating all the information from v-nodes. In the original version of soft belief propagation algorithm, c-nodes must apply multiple *arctanh* operations in order to calculate responses for v-nodes in the iteration process; it is known that basic algorithm can be modified to reduce the implementation complexity of the decoder. The basic idea is to replace all *arctanh* operations with approximations using tables or reduced complexity operations. Multiple approaches have been proposed as in [11] where a normalization factor based on calculating the original equation in the first iteration is proposed. These results are stored in a look up table and used to calculate a scaling factor for future values obtained by approximation. The proposed technique is no effective since it tries to correct an approximation using other approximation. [12] proposes a 2D correction, it means to apply normalization in c-nodes and v-nodes using parallel differential optimization, a search technique featured by evolution of variable vectors. In this approach, all input vectors must be stored for improving the cost function value with the consequent memory and power spend. [13] proposes to calculate correction factors in every iteration based on a density evolution function getting a very close result to the original min-sum algorithm. [13] presents simulation results in very short codes which limit its proposal to simulation environments. Finding a low complexity and effective normalization scheme is an important research line since it could reduce the total decoder complexity without diminishing its precision. In this point, minimize overhead, power consumption and use of memory resources are main targets.

V-nodes are responsible for receiving priori data from the channel, these data are the only known information before the decoder begins to work. When the decoder is iterating, v-nodes must decide if the resultant data word is a valid word (based on a vector-matrix multiplication), if it is true, v-nodes module generates a master signal to finish all the decoding process, sends the resultant data word to the system output and requests the next data word to start the process again. Because complexity of variable nodes is low compared with complexity of c-nodes, the most of works are centered in c-nodes; however, it is possible to distribute some of the performed processing in c-nodes or inclusive to improve effectiveness of the decoding processes including data normalization in v-nodes as proposed in [12]. It is possible to include other processes, too. For example the strategy for early termination (SET) [5] which allows to finish the decoding process when it is clear that a solution will not be reached. SET improves considerably the total decoding time. Other processes as direct memory access could be performed in v-nodes for diminishing the use of communication lines with c-nodes. Similar proposals could be a way for improving the final performance of the LDPC decoder since they allow to decentralize the processing load from c-nodes.

3.2 Memory data access

LDPC decoding process requires a high amount of information then, data access could be a bottleneck. In hardware implementations, memory requirements are directly related to the number of nodes to be processed in parallel and are determined by the data amount to be transferred between processing nodes. Basically, level of parallelism is limited by memory capacities due to the difficulty that represents accessing multiple data in the same clock cycle [6]. In general, the final decision about the order in which these data are accessed is based on the experience of the designer and there are no formal mechanisms for ensuring that any proposed solution is optimum. In reconfigurable implementations, memory access is an important challenge since nodes must access memory banks in a flexible fashion.

A possible option is to apply automatic parallelization techniques to the decoding algorithm. The main objective of automatic parallelization is to convert a sequential program into a parallel version that can directly run on multiple processing elements without altering the original semantics of the program. Automatic parallelization could be applied in hardware design too for modeling processor arrays for a sequential algorithm as in LDPC decoding. The idea of using modeling techniques for parallelization purposes is not new; however, it has not been applied in LDPC decoders design or in designing reconfigurable architectures. Polyhedral model applied to architectures design is a powerful tool for generating flexible models as well as exploring possible designs for processor units. Actually, research is required for exploring full capacities of applying this technique to reconfigurable architectures and specifically to LDPC decoders design.

3.3 Routing Network

Routing network responsibilities include to provide data in the correct form and time to v-nodes, c-nodes and to send the corresponding temporal data to memory banks. In a simple view, this looks like a simple function; however, when this network must support a reconfigurable system this process becomes a very complex task [4]. In order to facilitate the work of the routing network, an efficient design for v-nodes and c-nodes must be provided. Traditional approaches for building reconfigurable network includes:

1. Bus Based: Single / Multiple.
2. Switch based: SS (Single Stage) / MS (Multi Stage) / Cross Bar.

LDPC codes present better performance if the distribution of edges between nodes is random (non-structured), in other words there is no regular communication pattern, reason why still it is necessary to design an effective communication scheme for no regular patterns of communication. [3] presents how to address the variable block size and multi rate decoder hardware complexity that stems from the irregular LDPC codes, it is achieved using a barrel shifter network (switch based). The most of authors, present direct communication between nodes and memory banks which forces to use high amount of total area as memory resources (until 75%) as in [7].

It is shown in [4] that the overall performance is impacted by a limited bandwidth, reason why volume and format of data between nodes is a key aspect in any LDPC decoder design. Indeed, the correct data format could be the best option for optimizing use of network resources independently of its reconfigurability capacities. This is a poor worked research line.

In [4], throughput values obtained with multi-network insertion are improved by at least a factor 4 compared to the base case due to imposed latency for using a single communication bus. Single bus is the bottleneck when a wide range of standards are supported as in [7] where Viterbi, Turbo codes and LDPC codes are implemented based on share memory.

[4] shows the trade-of between a high level of generality versus poor results in key aspects as throughput and power consumption. An efficient network, able to support reconfigurability process in non-structures codes and to provide enough performance for most demanding applications is a challenge approached by multiple researchers and in continuous development.

4 Conclusions

Actually, there are multiple implementations for LDPC decoders including improved features. However, it is necessary to provide more flexible and efficient architectures in order to face future requirements including the ability of reconfiguration. Algorithmic improvements are an important research field but in general, it moves forward slowly reason why new architectural approaches could accelerate the development of future-proof radios. In this context, it is possible that modeling methodologies as the polyedral model provide the required framework to achieve optimum architectural solutions. Such solutions will satisfy requirements as high performance, low power consumption and improved flexibility.

References

1. Mitola III, Joseph. Software Radio Architecture. Object-Oriented Approaches to Wireless Systems Engineering. Wiley, 2000.
2. J. Chen and M.P.C. Fossorier. Near optimum universal belief propagation based decoding of low-density parity-check codes. Transactions on Communications, 50:406414, March 2002.
3. Y. Sun, M. Karkooti and J. R. Cavallaro, ”VLSI Decoder Architecture for High Throughput, Variable Block-Size and Multi-Rate LDPC Codes”. IEEE International symposium on Circuits and Systems (ISCAS 2007).
4. Scarpellino Michel; Masera Guido. Reconfigurable Architecture for LDPC and Turbo Decoding: A NoC Case Study, in Proc. IEEE 10th International Symposium

on Spread Spectrum Techniques and Applications (ISSSTA '08), Page(s):671-676, Aug. 2008.

5. Bing Li, Duo-jin Chen, Jun Xiong. Implementation of a multi-rate and multi-size LDPC decoder. IEICE Electronics Express, Vol. 6, No.21, 1509-1515. 2009.
6. T. Mohsenin, D. Truong, and B. Baas, An improved Split-Row Threshold decoding algorithm for LDPC codes, IEEE International Conference on Communications, June 2009.
7. Alles M., Vogt T., Wehn, N., "FlexiChaP: A reconfigurable ASIP for convolutional, turbo, and LDPC code decoding", 5th International Symposium on Turbo Codes and Related Topics, Switzerland 2008.
8. Kienle, F., Brack, T., Wehn, N. "A synthesizable IP core for DVB-S2 LDPC code decoding", Design, Automation and Test in Europe Conference and Exposition, Munich, Germany, 7-11 March 2005.
9. J. Zhang, M. Fossorier, D. Gu, and J. Zhang, Improved min-sum decoding of LDPC codes using 2-dimensional normalization, in Proc. Globecom 2005, IEEE Global Telecommunications Conference, St. Louis, Missouri, USA, 2005.
10. J. Chen, R.M. Tanner, C. Jones, and Y. Li, Improved min-sum decoding algorithms for irregular LDPC codes, in Proc. 2005 IEEE International Symposium on Information Theory, Adelaide, Australia, 2005, pp. 449453.
11. Kim, S.; Sobelman, G. E.; Lee, H., "Adaptive quantization in min-sum based irregular LDPC decoder," Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on, pp. 536-539, May, 2008.
12. J. Zhang, M. Fossorier, D. Gu, and J. Zhang, Improved min-sum decoding of LDPC codes using 2-dimensional normalization, in Proc. Globecom 2005, IEEE Global Telecommunications Conference, St. Louis, Missouri, USA, 2005.
13. J. Chen, R.M. Tanner, C. Jones, and Y. Li, Improved min-sum decoding algorithms for irregular LDPC codes, in Proc. 2005 IEEE International Symposium on Information Theory, Adelaide, Australia, 2005, pp. 449453.