SUM-SUBTRACT FIXED POINT LDPC DECODER

L. ARNONE†, C. GAYOSO†, C. GONZÁLEZ† and J. CASTIÑEIRA‡

†*Laboratorio de Componentes Electrónicos, Facultad de Ingeniería, UNMDP, Argentina.*
leoarn@fi.mdp.edu.ar
‡ *Laboratorio de Comunicaciones, Facultad de Ingeniería, UNMDP, Argentina*
casti@intec.edu.ar

***Abstract*— In this paper a low complexity logarithmic decoder for a LDPC code is presented. The performance of this decoding algorithm is similar to the original decoding algorithm´s, introduced by D. J. C. MacKay and R. M. Neal. It is a simplified algorithm that can be easily implemented on programmable logic technology such as FPGA devices because of its use of only additions and subtractions, avoiding the use of quotients and products, and of float point arithmetic. The algorithm yields a very low complexity programmable logic implementation of a LDPC decoder with an excellent BER performance.**

***Keywords*— low density parity check codes, decoding, BER performance, look-up tables**

I. INTRODUCTION

A method for decoding Low Density Parity Check Codes (LDPC) is the sum-product algorithm proposed by Gallager (Gallager, 1962). Quotients and products involved in this algorithm make difficult the implementation of optimal LDPC decoders on low complexity programmable logic.

In this paper we propose a very low complexity sum-subtract fixed point decoding algorithm for LDPC codes. This algorithm also uses two look-up tables.

A comparison is done between the BER performance of the proposed decoding algorithm, and the BER performance of Gallager's sum-product decoding algorithm (Gallager, 1962) for a given LDPC code.

Results show that there is no significant difference in BER performance between the optimal and the proposed algorithm. The proposed algorithm is characterized by a very low complexity implementation, thus becoming a better alternative for its programmable logic implementation than the traditional sum-product algorithm.

This paper is organized as follows: Section II shows the main aspects of a LDPC decoder. Section III introduces the proposed algorithm. Section IV is related to the look-up tables utilized in the proposed algorithm. Section V is devoted to a comparative complexity analysis for these two LDPC decoding algorithms. Section VI shows BER performance results, and finally Section VII deals with the conclusions.

II. LDPC CODES

LDPC codes (Gallager, 1962) are a powerful class of linear block codes characterized by a parity check matrix $\mathbf{H}$, which fits the condition $\mathbf{H} \circ \mathbf{x} = \mathbf{0}$ for any codeword $\mathbf{x}$. A LDPC decoder is essentially a decoding algorithm (MacKay and Neal, 1997; MacKay, 1999) designed for finding a codeword $\hat{\mathbf{d}}$ (an estimate of the codeword $\mathbf{x}$), able to fit the condition:

$$\mathbf{H} \circ \hat{\mathbf{d}} = \mathbf{0} \qquad (1)$$

The LDPC decoding algorithm is described over a bipartite graph depicted considering the relationship between the symbol nodes $d(j)$, which represent the bits or symbols of the code vector $\boldsymbol{x}$, and the parity check nodes $f(i)$, which represent the parity equations described in matrix $\mathbf{H}$. In this iterative process, each symbol node $d(j)$ sends to a parity check node $f(i)$ the estimation q_{ij}^{x} that this node generates with the information provided by all others parity check nodes connected to it, based on the fact that the parity check node j is in state x.

Then, each parity check node $f(i)$ sends the estimation r_{ij}^{x} to each symbol node $d(j)$ generated with the information provided by the other symbol nodes connected to it, based on the fact that the parity check node i condition is satisfied, if the symbol node $d(j)$ is in state x. This is an iterative process in which information is interchanged between these two types of nodes. This iterative process is stopped when the condition described by Eq. (1) is satisfied. In this case the corresponding decoded codeword is considered a valid codeword. Otherwise, the decoding algorithm stops after a given number of iterations are performed. In this case the decoded word may or may not be a codeword.

III. THE PROPOSED ALGORITHM

The bases of the LDPC decoding algorithm are described in (MacKay and Neal, 1997). The proposed simplification makes this algorithm operate using only additions and subtractions. This simplification makes use of a logarithmic version of the calculations involved in the original algorithm. The proposed algorithm is procedure based on the fact that a given number z,