Banyan Switch Applied for LDPC Decoder FPGA Implementation

Wojciech Sułek [*]

[*] *Silesian University of Technology, Faculty of Automatic Control,
Electronics and Computer Science, Akademicka 16, 44-100 Gliwice
(e-mail: wsulek@polsl.pl)*

Abstract: Low-Density Parity-Check codes are one of the best modern error-correcting codes due to their excellent error-correcting performance and highly parallel decoding scheme. This article concerns the hardware iterative decoder for a subclass of LDPC codes that are implementation oriented, known also as Architecture Aware LDPC. The parameterizable decoder has been designed in the form of synthesizable VHDL description. Implementation in Xilinx FPGA devices achieves throughput nearly 100Mb/s. Significant part of the decoder area is occupied by Configurable Interconnection Network. The network consists of a set of multiplexers that propagate the data from memory to the computation units. Behavioral description of the interconnection network gives quite poor synthesis results: decoder area is large and exponentially dependent on the number of inputs / outputs. Instead of straightforward behavioral description, the switching network can be described structurally making use of ideas known from theory of telecommunication interconnection networks: Benes or Banyan switches. In this article I present in detail the interconnection network implementation based on Banyan switch with additional multiplexer stage to enable non-power-of-2 numbers of outputs. Comparison of synthesis results for the network obtained by synthesis of behavioral description as well as the Banyan structural description shows significant decrease of decoder area in the second case.

Keywords: Error-correcting codes, LDPC codes, Decoders, Interconnection networks, Banyan networks

1. INTRODUCTION

Outstanding error correction performance and highly parallel iterative decoding algorithms have made Low-Density Parity Check (LDPC) codes (MacKay (1999)) the most powerful error correcting codes for reliable high speed communication applications. They have been recently adopted for variety of industrial standards, e.g. IEEE (2006), ETSI (2009). Besides all their desirable properties, one characteristic of LDPC codes, namely the randomness of parity-check matrix structure, makes implementation of LDPC decoders a difficult task as it leads to complex interconnect wiring for practical codes, and hence to a large demands for hardware resources in a decoder implementation.

The typical LDPC decoding procedure is the message passing algorithm, also known as belief propagation (BP) algorithm, that implements the iterative updating of messages (i.e. beliefs for codeword bit values) based on the parity-check equations for the particular code. These parity-check equations can be represented graphically by a bipartite Tanner graph. Thus the Tanner graph indicates the message passing scheme.

Several LDPC decoder architectures have been proposed to solve the problem of efficient message delivery between the computation units. A fully parallel architecture (e.g. Blanksby and Howland (2002)) exploits the benefit of parallel decoding schemes. However, the implementation complexity is very high and impractical for codes of larger blocks. On the other side, serial architectures where the single processing unit is shared for area efficiency are too slow for most applications. Thus partially parallel architectures, with some limited number of processing units, allow proper cost-performance tradeoffs, see e. g. Zhong and Zhang (2005); Lee and Ryu (2008). It has been recognized that the partially parallel decoder architectures can be accomplished well for some subclass of codes, with structured parity check matrix, known as Architecture-Aware LDPC (Mansour and Shanbhag (2003)) or VLSI-Oriented (Zhong and Zhang (2005)).

In Architecture-Aware LDPC (AA-LDPC) codes, the parity check matrix is composed of permutation submatrices that determine the interconnection network configuration in the consecutive subiterations. In this paper I briefly present partially parallel AA-LDPC decoder that has been implemented in the form of synthesizable VHDL description (Sułek (2008)). The target hardware platforms for the decoder are FPGA devices. Then I focus on the construction of configurable interconnection network, which constitutes a significant part of the required FPGA decoder area.

Behavioral description of the interconnection network leads to quick increase of the required number of resources with the number of network inputs / outputs. Significantly more effective use of resources can be achieved with structural description of the network in the form of well known Benes networks that has been adapted by some researchers (Tang et al. (2006), Lin et al. (2009)). However, if the submatrices of the parity check matrix are in the form of cyclic shifts of the identity matrix, then the switch can be constructed with even more ares efficient Banyan switch (e.g. Wu and Feng (1980)). In this paper I present the details of the Banyan switch implementation for LDPC configurable interconnection network and the synthesis results of the implemented network for the Xilinx VirtexII devices with comparison to the results obtained for the behavioral description of the network.

2. LDPC CODES BASICS

Low-density parity-check codes are a sub-class of linear block codes that are defined by a very sparse parity check matrix $\mathbf{H}_{M\times N}$, MacKay (1999). Encoding process for a linear code denoted as $\mathcal{C}(N, N-M)$ adds M redundant bits to the information vector $\mathbf{u} = \{u_1, u_2, \ldots, u_{N-M}\}$ to form the code vector $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$. In the decoder, the vector $\mathbf{x}$ is recognized as a correct codeword ($\mathbf{x} \in \mathcal{C}$) if and only if the parity check condition $\mathbf{Hx}^{\mathrm{T}} = \mathbf{0}$ is satisfied (in $GF(2)$ field). If the codeword has been corrupted, the decoding algorithm tries to correct errors. The coderate $R = (N-M)/N$ characterizes the amount of redundancy in the code.

LDPC codes are often defined by an alternative to parity check matrix representation: a bipartite Tanner graph, where one set of nodes represents data symbols (bits), also known as bit nodes, and the other set represents parity check constraints (check nodes). Each edge in the Tanner graph corresponds to a '1' in the parity check matrix $\mathbf{H}$. Figure 1 shows the Tanner graph and the parity check matrix for a linear code with $N = 7$ bit nodes and $M = 3$ check nodes. Actually this graph represents Hamming $\mathcal{C}(7,4)$ code, which is not an LDPC code, but it is unessential for Tanner graph definition.

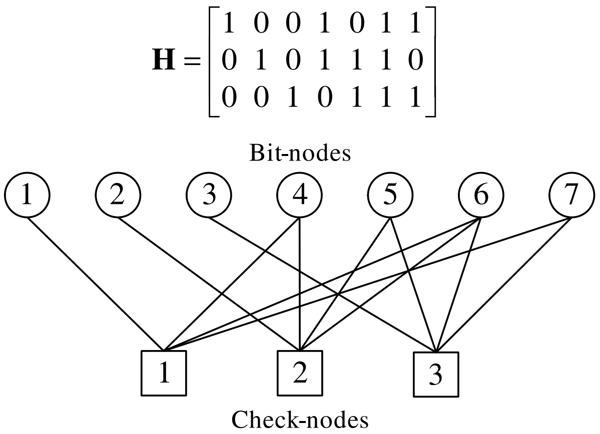

Fig. 1. Tanner graph

The Tanner graph visualizes iterative message passing algorithm used for decoding. Commonly used algorithms are performed by exchanging messages (beliefs) between bit nodes and check nodes through the edges in both directions. Each node of the graph represents computation of updated beliefs and edges indicate the way of message passing. In the case of LLR-BP algorithm (Log-Likelihood Ratio Belief Propagation) messages are log-likelihood ratio of beliefs, then operations are sums (bit nodes) and sums of some nonlinear functions of messages (check nodes). Inputs to the algorithm are measures of reliability of the received data based on channel observations (received channel soft values), which are in the form of log-likelihood ratios of probabilities. They are called intrinsic channel reliability values, denoted as δ_n for nth bit of the codeword:

$$\delta_n = \log\left[\frac{P(x_n = 0|y_n)}{P(x_n = 1|y_n)}\right] \qquad (1)$$

where x_n is the transmitted bit and y_n is the received "soft" value. Details of the iterative message passing decoding algorithms will be omitted here, because they are available in reach literature, e.g. MacKay (1999); Lin and Costello (2004); Zhao et al. (2005).

Hardware decoder that will be briefly presented in the next section is based on TDMP (Turbo-Decoding Message-Passing) version of decoding scheme, see Mansour (2006). TDMP algorithm uses a modified message passing scheme, where $\mathcal{C}$ is considered as the concatenation of some number of supercodes, which means the set of rows of $\mathbf{H}$ is virtually partitioned into a number of subsets. Each subset constitutes a supercode. A word is a correct codeword if it belongs to all constituent supercodes. In the TDMP algorithm implementation, only one kind of computation units is needed that is called SISO unit (Soft-Input Soft-Output).

2.1 Architecture-Aware LDPC Codes

As is well known (e.g. Mansour and Shanbhag (2003); Zhong and Zhang (2005)), efficient partially-parallel decoder implementation is possible for parity-check matrices with certain constraints on their form. The main building blocks of partially-parallel decoder are message memory and some number of computation units. In order to suitably organize message memory access and eliminate conflicts, the code graph has to posses some specific properties. The set of bit nodes of the graph is partitioned into L subsets of P nodes and similarly the set of check nodes is partitioned into D subsets of P nodes. Here P is the number of computation units used for implementation. Edges in the graph should be organized such that one of the following conditions is satisfied for every combination of bit node subset and check node subset:

- each bit node from the subset is adjacent to exactly one check node in a check node subset,
- each bit node in the subset is not adjacent to any check node in a subset.

Then single memory word may consist of P messages that are delivered to P computing modules in a single clock period by configurable interleaver that shuffles the fragments of the memory word according to the local graph structure of the code.

With such code-graph organization, a parity check matrix is similar to the one shown in equation (2):