Efficient Hardware Implementation of Probabilistic Gradient Descent Bit-Flipping

Abstract:

This deals with the hardware implementation of the recently introduced Probabilistic Gradient-Descent Bit-Flipping (PGDBF) decoder. The PGDBF is a new type of hard-decision decoder for Low-Density Parity-Check (LDPC) code, with improved error correction performance thanks to the introduction of deliberate random perturbation in the computing units. In the PGDBF, the random perturbation operates during the bit-flipping step, with the objective to avoid the attraction of so-called trapping-sets of the LDPC code. In this paper, we propose an efficient hardware architecture which minimizes the resource overhead needed to implement the random perturbations of the PGDBF. Our architecture is based on the use of a Short Random Sequence (SRS) that is duplicated to fully apply the PGDBF decoding rules, and on an optimization of the maximum finder unit. The generation of good SRS is crucial to maintain the outstanding decoding performance of PGDBF, and we propose two different methods with equivalent hardware overheads, but with different behaviors on different LDPC codes. Our designs show that the improved PGDBF performance gains can be obtained with a very small additional complexity, therefore providing a competitive hard-decision LDPC decoding solution for current standards.

Existing system

The GDBF algorithm is derived from a gradient descent formulation and its principle consists of finding the most suitable bits to be flipped in order to maximize a pre-defined objective function. The GDBF algorithm showed an error correction capability superior to most known BF algorithms while still keeping the hardware implementation simplicity. A very promising generalization of the GDBF is to incorporate a probabilistic feature in the flipping step, inspired from the probabilistic BF algorithms of . In the PGDBF decoder, the bits that satisfy the gradient
condition are not flipped by default, but instead, only a randomly chosen fraction $p_0$ of them are flipped. Interestingly, this small modification of the GDBF algorithm led to a large performance improvement, with error correction capability approaching the soft-decision message passing decoders.

**Proposed system**

An efficient hardware (HW) implementation of the PGDBF decoder, which minimizes the resource overhead needed to implement the random perturbations of the PGDBF. We restrict our work to decoders for regular LDPC codes used over the binary symmetric channel (BSC). Although many wireless communication standards use irregular LDPC codes, regular LDPC codes could still be interesting for practical applications, such as fiber optics transmissions, satellite communications using short frames or storage applications.

**Applications**

1) Communications  
2) Digital signal processing

**Advantages**

Area, delay and power reduced

**System Configuration:**

In the hardware part a normal computer where Xilinx ISE 14.3 software can be easily operated is required, i.e., with a minimum system configuration

**HARDWARE REQUIREMENT**

<table>
<thead>
<tr>
<th>Processor</th>
<th>Pentium –III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>1.1 GHz</td>
</tr>
</tbody>
</table>
SOFTWARE REQUIREMENTS


- Front End: Modelsim 6.3 for Debugging and Xilinx 14.3 for Synthesis and Hardware Implementation

This software’s where Verilog source code can be used for design implementation.