NB-LDPC Code Decoder Simulator User Guide

Author: Eric Reiss

Introduction

The purpose of this project is to provide a framework for developing and testing non-binary low-density parity-check (NB-LDPC) code decoding algorithms. This guide provides the information necessary to set up a project, implement a decoding algorithm, and obtain the results. To successfully use this tool, the user should have a working knowledge of communication theory, NB-LDPC codes, and C++.

The sections discussed in this guide are:

- Tool Description
- Required Equipment and Software
- Safety Considerations
- Directions
- Troubleshooting
- Conclusion

Tool Description

This tool is implemented using SystemC, which is an extension of C++. Building the tool is handled by the CMake toolchain. The repository is organized by directories containing the files necessary to build and simulate a decoder. The directories included are codes, inc, opt, and src. A description of each is included below. The simulator comes with an implementation of the Belief Propagtion algorithm, which is commonly used for binary LDPC code decoders.

SystemC

SystemC is a library that extends C++ and adds hardware simulation capabilities. Classes in SystemC are analagous to modules or entities in Verilog or VHDL, respectively. Each class defines input and output ports and instantiates lower level classes within it. Theses instantiated classes are connected, similar to an HDL. Each class defines a behavior function and sensitivity list. At the edge of a sensitive signal, the behavior function is called. This is analagous to an always block or process in Verilog or VHDL, respectively.

Belief Propagation

TODO: Describe the belief propagation algorithm. Include Tanner graph diagram

Directories

The purpose of this subsection is provide a description of the organization of the repository.

NB-LDPC This is the top level directory of the the project and contains the files needed to start the build process with the CMake toolchain. The important files are described below.

- **COPYRIGHT** This file contains the copyright for this software.
- **Makefile** The Makefile is the script that starts the CMake tool chain and builds the project.
- **systemc.mk** This file contains configuration details specific to SystemC
- **README.md** The README contains this document in Markdown.

codes The codes directory contains the alist files of common LDPC matrices. The alist format was developed by David Mackay as a way to efficently sparse matrices. More information can be found on David Mackay's website (link). The codes directory contains alist files for the binary 802.11n and PegReg codes as well alist files for GF(4) and GF(8) NB-LDPC codes. The alist files provided by David Mackays website (code file downloads)

inc The inc folder contains the header files used in the simulation. The header files contain most of the code that is specific to a decoder implementation. The following files are included with the repository.

- **alist.h** Header file that defines the functions used to process the alist file
- **decoder.h** Header file that defines the decoder module
- **LDPC_testbench.h** Header file that tefines the testbench module
- **ldpcsim.h** Header file for the top level SystemC file, this is not a module
- **nodes.h** Header file that defines the symbol node and check node modules
- **nrutil.h** Header file containing numerical methods
- **r.h** Header file that defines useful helper functions, written by David Mackay
- **rand.h** Header file that defines randomization constants and methods, written by David Mackay
- **sc_vector.h** SystemC library header file that allows vector inputs and outputs
- **sc_vector.ipp** Optimized implementatoin of sc_vector methods

src The src folder contains all the source cpp files that are compiled by the build tool. The SystemC source files define all the module connections and generate the inputs. The following files are included in the src folder.

- **alist.cpp** Processes the alist files and creates the code struct
- **ldpcsim.cpp** Top level SystemC file that instantiates the testbench and decoder modules

- **nrutil.cpp** File provided by David Mackay with useful helper functions
- **r.cpp** File provided by David Mackay that contains randomization functions

Required Equipment and Software

To effectively use this tool, the following hardware is required:

- Computer with at least 8GB RAM

The following software is required:

- Linux OS 64-bit
- GNU Make v4.3 or later
- GNU Compiler v11.4.0 or later
- CMake v3.22.1 or later (CMake Website)
- SystemC v3.0.0 or later (SystemC GitHub page)
- IT++ Library v4.3.0 or later (IT++ Website)

Safety Considerations

TODO: Figure out what safety considerations there are

Directions

TODO: Talk about installing necessary tools, building project, runtime constants, interpreting constants

Troubleshooting

TODO: Talk about checking library file installation location, Makefile parameters, and C++ debugging tools.

Conclusion

TODO: Conclude