Design and FPGA Implementation of LDPC Decoder using High Level Modeling for WSNs

Mallesha B. Y

Abstract*—Low density parity check (LDPC) codes are error-correcting codes that offer huge advantages in terms of coding gain, throughput and power dissipation in digital communication systems. Error correction algorithms are often implemented in hardware for fast processing to meet the real-time needs of communication systems. However, traditional hardware implementation of LDPC decoders require large amount of resources, rendering them unsuitable for use in energy constrained sensor nodes of wireless sensor networks (WSN). This paper investigates the use of short-length LDPC codes for error correction in WSN. It presents the LDPC decoder designs, implementation, resource requirement and power consumption to judge their suitability for use in the sensor nodes of WSN. Due to the complex interconnections among the variable and check nodes of LDPC decoders, it is very time consuming to use traditional hardware description language (HDL) based approach to design these decoders. This paper presents an efficient automated high-level approach to designing LDPC decoders using a collection of high-level modeling tools. The automated high-level design methodology provides a complete design flow to quickly and automatically generate, test and investigate the optimum (length) LDPC codes for wireless sensor networks to satisfy the energy constraints while providing acceptable bit-error-rate performance.

***Index Terms*—Error Correction Coding; Wireless Communication; Wireless Sensor Networks; Digital System;**

I. INTRODUCTION

Low Density Parity-Check (LDPC) codes [1, 2] are known as the most powerful forward error correction codes with a bit-error-rate (BER) performance closed to the Shannon limit. LDPC codes have been proved to have better performance and several advantages over other error correction codes such as Turbo codes, Hamming codes, Reed-Muller and Reed-Solomon codes [2]. Because of excellent BER, LDPC-codes are extensively used in standards such as WiMAX, 10Gigabit Ethernet (10GBaseT), digital video broadcasting (DVB-S2) and expected to be part of many future standards [3, 4]. Although the decoding algorithm of LDPC is simple, hardware implementation faces several significant challenges. One of the challenges in implementation of fully parallel LDPC decoder is the complexity of the interconnections between the nodes inside the decoder [5]. Especially when the LDPC matrix becomes large, it is almost impossible and time consuming to manually connect and check the connections. In this paper, an automated high-level design methodology is introduced. We propose a design methodology that supports programmable logic design starting from high-level modeling all the way up to FPGA implementation using a collection of high-level modeling tools. The methodology has been used to design and implement LDPC decoders of various code lengths on FPGAs. The simulation and FPGA implementation results obtained are then used to determine suitable LDPC decoders for Wireless Sensor Networks (WSN), which consist of severely resource constrained sensor nodes.

In recent years, WSNs have attracted significant research interests [6-8]. A WSN can be defined as a network of a large number of spatially distributed, small, low cost and low power nodes, which can sense the environment and wirelessly communicate the information gathered to other nodes. The collected information is forwarded, normally via multiple hops, to a sink (or controller or monitor) node that uses the information locally or transmits it to other networks (e.g., Internet) through a gateway. WSNs are normally comprised of scalar sensors capable of measuring physical phenomenon such as temperature, pressure, light intensity, humidity etc. [9]. The abovementioned applications do not have a high bandwidth requirement and are delay tolerant. Recently, several research works have been reported to add small sized and low-power CMOS cameras and microphones to the sensor nodes. Such Wireless Multimedia Sensor Network (WMSN), with the ability to gather multimedia information from the surrounding environment, is providing the impetus for extending the capabilities WSNs for many new applications such as advanced environmental monitoring, advanced health care delivery, traffic avoidance, fire prevention and monitoring, object tracking etc. However, in WMSN, with the large volume of multimedia data generated by the sensor nodes, both processing and transmission of data leads to higher levels of energy consumption. Energy consumption in transmitting large volume of multimedia data can be reduced by using energy-efficient and reliable transmission protocols [10] or reducing the amount of multimedia data [11]. Beside these approaches, efficient error correction decoders can be used to reduce the energy required for communication of the multimedia information. However, the amount of energy spent to transmit the redundant information required for error correction and the energy used to perform error correction should be less than the energy saved at the transmitter side for retransmission of erroneous information [12]. Several codes have been investigated for error correction in WSN, including Reed–Solomon codes, convolution codes, turbo codes and LDPC codes [13, 14]. Some preliminary results in [12, 15] suggest that LDPC codes are good candidates for WSN applications as they feature a significant coding gain compared with other codes. LDPC codes are known to achieve nearly the Shannon limit with long code length. However, parallel decoders for long LDPC codes require large hardware and energy consumption and therefore partially parallel decoders were introduced [16].

Revised Version Manuscript Received on September 19, 2015.

Mallesha B. Y, P G Student, Visvesvaraya Technological University Extension Centre, UTL Technologies Ltd., Yeshwantpur, Bangalore, India.

In WSN, the data exchanged between sensor nodes is usually small [10]. This raises the prospect of using short LDPC codes for error correction in WSN for without compromising the *bit-error-rate* (BER) performance significantly. Most of the previous works proposing error correction codes for WSNs assume that networks contain only two types of nodes: sensing nodes and base stations. Sensing nodes feature lower computational capabilities and lower available energy than base stations. Thus, sensing nodes send coded information to a central node which performs the decoding operations. In this paper, we investigate the possibility of implementing short length LDPC codes in WSNs where sensor nodes can both encode/transmit and receive/decode information. We show that that LDPC codes with small block length are adequate for typical throughput and data transmission requirements of WSNs.

II. AUTOMATED HIGH-LEVEL DESIGN METHODOLOGY

The proposed automated high-level design methodology provides a complete design flow to automatically generate and test complex LDPC decoders. As shown in Fig. 2, the methodology has two main parts, namely, automatic generation of decoder models and automatic testing of decoders (presented in Part B of this section).

A. Automatic Generation of LDPC Models and HDL Codes. The automatic LDPC model generation flow shown in Fig. 2 requires two inputs: LDPC library and LDPC matrix. The LDPC library contains two basic hand designed modules, check node (C-node) and variable node (V-node). The LDPC matrix can be preconfigured to any size for (3, 6)-regular LDPC code.

1). Design of Check Node and Variable Node

The C-node has been designed in Simulink using builtin Simulink library blocks. It performs the operation given in (2). The main function of the C-node is to find the minimum of all the inputs to the C-node and to perform parity checks. Several approaches to design the C-node have been evaluated for hardware requirement and speed to choose the most optimized design for the automated design flow. For LDPC (3, 6) code, every C node has six inputs. First, the C-node was designed using (2), by comparing every set of five inputs separately, to find the minimum. In this approach the hardware requirement for the C-node is high due to the same comparisons being made multiple times [19]. An optimized find-minimum-function is presented in Fig. 3. In this design, a module called find3min is used to find the minimum of 3 sets of 4 inputs. For example, Min1234 is the minimum of input set {1, 2, 3, 4}. Each output of the find3min block is then compared with one more input to find the minimum of five inputs. As the outputs of every fin3min block are reused, this approach reduces the amount of hardware resources required to implement the find-minimum-function. Simulink model of the C-node is shown in Fig. 1.

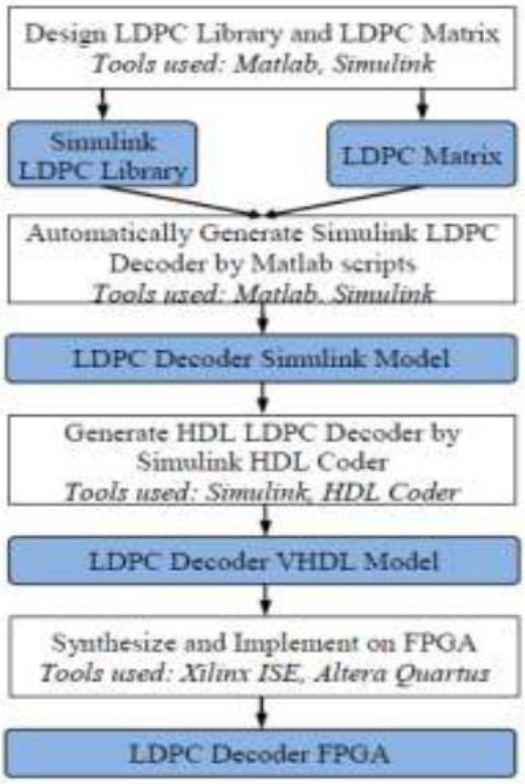


Fig 1. Automated LDPC design flow

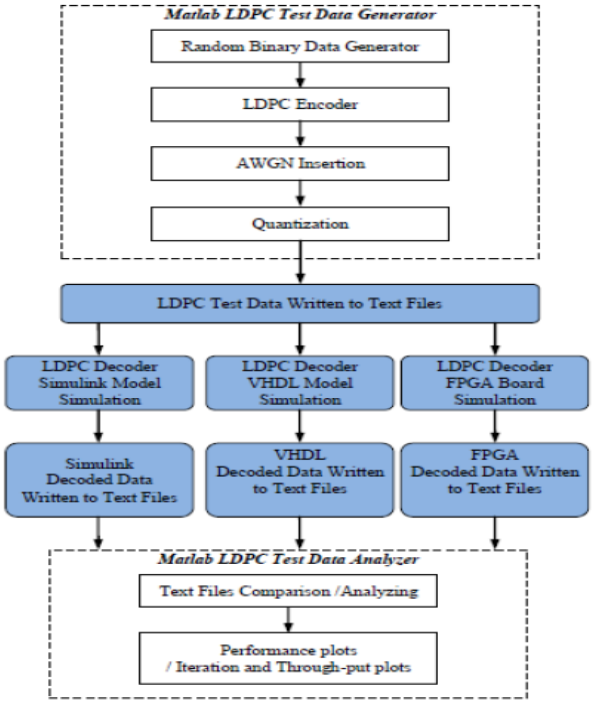
2). Auto-generation of Decoder Models

The auto-generation of decoder models is accomplished using Matlab scripts. The Matlab scripts read the LDPC matrix and automatically instantiate the V-nodes and C-nodes to build the corresponding decoder model in Simulink. They also connect the V-nodes and C-nodes according to the LDPC matrix. The routing process is very simple but effective. Each input and output of the V-nodes and C-nodes are marked with a Simulink routing label. The routing labels are specified by the Matlab scripts to build the connections between nodes as required by the LDPC matrix. If a routing label of a V-node input has the same name as that of the routing label of a C-node output then Simulink will understand these two ports to be connected to each other. Once a Simulink decoder model is generated by the Matlab script, it can be used to automatically generate a HDL model using the Simulink HDL Coder tool. Either VHDL or Verilog can be chosen as the target HDL. The generated HDL code is vendor independent, and can be synthesized and implemented on most FPGAs.

III. AUTOMATIC TESTING STRATEGY

The proposed automatic model generation process is complemented with a comprehensive automatic testing strategy. It comprises an automatic process of generating and encoding test data, inserting noise, quantization and application of the final test data to the decoder under test. The Matlab and Simulink environments used in this highlevel design methodology provide an efficient and fast mechanism to build a full test system for the entire design.

The proposed testing strategy is comprehensive as it allows for testing the decoder at all possible levels of the hierarchy: Simulink model, HDL model and FPGA implementation. Fig.2 shows the proposed testing strategy.



IV. CONCLUSIONS

This paper has presented the designs and practical implementation results of short-length LDPC decoders for wireless sensor networks using a flexible automated methodology. FPGA test results have proved that the design methodology is efficient and error-free. The proposed methodology offers great advantages in terms of reduced design complexity, effort and time. It allows designers to quickly determine the trade-offs between shorter LDPC codes and shorter LLRs versus BER performance. The practical test results presented in the paper have demonstrated that short-length LDPC codes with small LLRs can be used for error correction at low power consumption while providing acceptable bit-error rate performance. The results of this study are therefore useful to determine optimum LDPC codes for low power applications such as wireless sensor networks.

REFERENCES

1. R. Gallager, "Low-density parity-check codes," IRE Transactions on Information Theory, vol. 8, pp. 21-28, 1962.
2. S. J. Johnson, "Introducing low-density parity-check codes," University of Newcastle, Australia, 2006.
3. M. Eroz, F. W. Sun, and L. N. Lee, "DVB-S2 low density parity check codes with near Shannon limit performance," International Journal of Satellite Communications and Networking, vol. 22, pp. 269-279, 2004.
4. T. Mohsenin and B. M. Baas, "Split-Row: A reduced complexity, high throughput LDPC decoder architecture," in International Conference on Computer Design, ICCD, San Jose, 2007, pp. 320-325.
5. Darabiha, A. C. Carusone, and F. R. Kschischang, "A bit-serial approximate min-sum LDPC decoder and FPGA implementation," in IEEE International Symposium onCircuits and Systems, ISCAS, Island of Kos, 2006.
6. D. Culler, D. Estrin, and M. Srivastava, "Overview of wireless sensor networks," IEEE Computer, Special Issue in Sensor Networks, vol. 37, pp. 41–49, 2004.
7. C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An Overview on Wireless Sensor Networks Technology and Evolution," Sensors, vol. 9, p. 6869, 2009.
8. M. Tubaishat and S. Madria, "Sensor networks: an overview," IEEE potentials, vol. 22, pp. 20-23, 2003.
9. Akyildiz, T. Melodia, and K. Chowdhury, "A survey on wireless multimedia sensor networks," Computer Networks, vol. 51, pp. 921-960, 2007.
10. S. M. Aziz and D. M. Pham, "Energy Efficient Image Transmission in Wireless Multimedia Sensor Networks," IEEE Communications Letters, vol. 17, pp. 1084 -1087, June 2013.
11. D. M. Pham and S. M. Aziz, "Object extraction scheme and protocol for energy efficient image communication over Wireless Sensor Networks," Computer Networks, Elsevier, Online, July 2013.
12. E. Sanchez, F. Gandino, B. Montrucchio, and M. Rebaudengo, "Increasing effective radiated power in wireless sensor networks with channel coding techniques," in International Conference on Electromagnetic in Advanced Applications, ICEAA, 2007, pp. 403-406.