Low Density Parity Check Code Implementation

Functional Description and System Block Diagram

By

Matthew Pregara and Zachary Saigh

Advisors: Dr. In Soo Ahn and Dr. Yufeng Lu

November 7, 2012

Introduction

In communication systems, forward error correction (FEC) codes have been widely used to battle data transmission errors caused by unreliable communications through a noisy channel. By adding extra bits to the end of message bits, a certain number of bit errors can be detected and corrected without frequent retransmission in case of data errors. Low density parity check (LDPC) is a powerful FEC coding scheme which can achieve good error performance under very low signal-to-noise ratios. [1] A communications system utilizing LDPC code is able to get very close to the channel capacity limit which was established by Claude Shannon in the 1940's. In addition, LDPC code has less complexity in the decoding process compared with other FEC codes. With the advance in computing power, it has been adopted in many high speed communication standards such as digital video broadcasting, WiMAX, 4G wireless systems, among others.

Functional Description

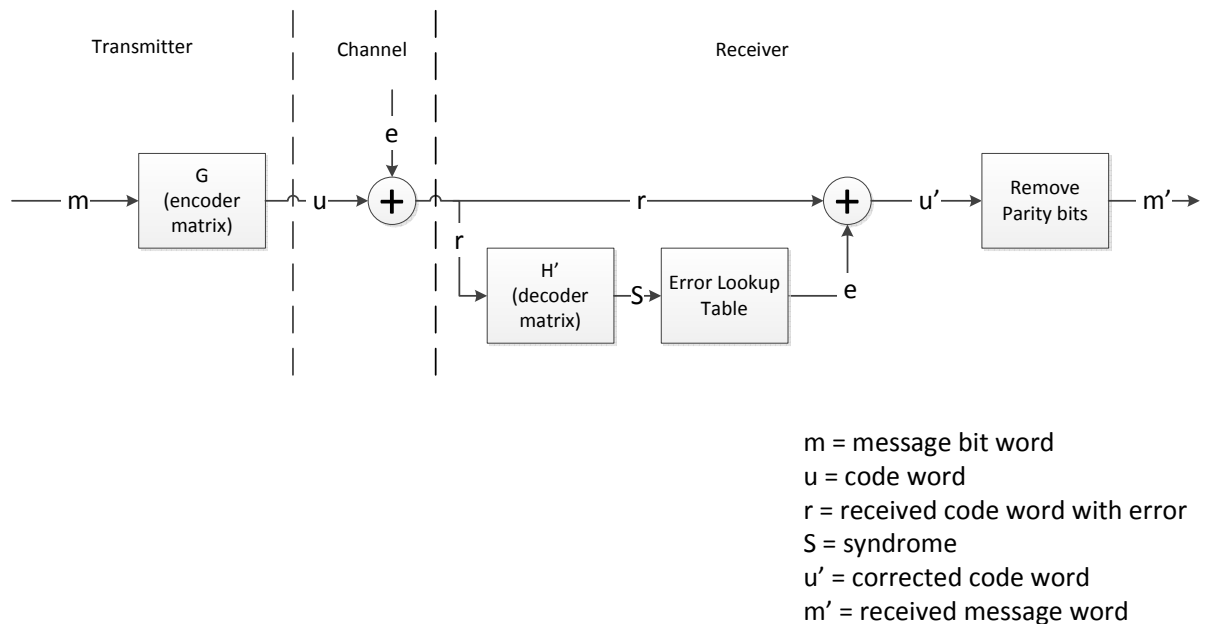A high level block diagram for a LDPC system is shown in Figure 1.

m = message bit word
u = code word
r = received code word with error
S = syndrome
u' = corrected code word
m' = received message word

Figure 1. High level block diagram for LDPC system

The system is described as follows. In the encoding process, an encoder matrix (**G**) is used to convert a message word to a code word (**U**). [2] The encoded message is then transmitted through a communication channel where multiple bit errors are introduced. In our study, modulation and demodulation are not considered. At the decoding side, a corrupted message (**r**) is received. The syndrome information (**S**) is obtained by using the decoder matrix (**H**). A look-up table listing error patterns vs syndromes is then used to determine the bit error pattern. When the bit error pattern is identified, it is used to correct the corrupted code word. Finally, the extra bit information (i.e., parity bits generated by the encoder matrix **G**) is removed and the result is the corrected message.

Project

The project can be divided into two parts. One is to construct a MATLAB/ Simulink model to demonstrate encoding and decoding processes of LDPC code. Another is to implement a scaled version of the LDPC decoding process using a Field Programmable Gate Array (FPGA) board. Two-stage block diagrams are shown in Figure 2 and Figure 3. A Simulink-based FPGA programming tool (Xilinx system generator) will be used for implementation.

It is not feasible to implement a LDPC code system for a large dimensional H matrix on an FPGA. Therefore, a relatively small H matrix and corresponding G matrix will be simulated and tested using MATLAB/Simulink. The simulated and verified LDPC code will then be implemented on an FPGA board. The performance of the implemented LDPC code system will be compared with simulation results. Specifications such as bit rate, channel bandwidth requirements, and error detection capability will be verified.
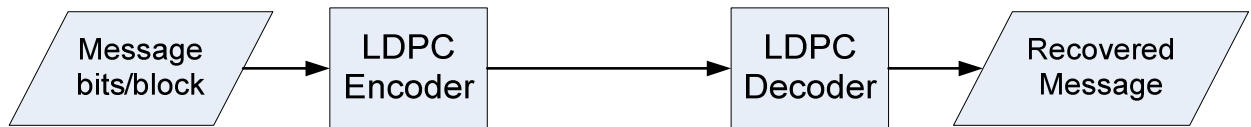
Figure 2. Design and test stage (MATLAB/Simulink)

Figure 3. Channel simulation and board implementation stage (Simulink/FPGA)

Functional Requirements

- Xillinx Spartan 3E board shall be used for implementation with the on board 50Mhz clock
- The decoder shall perform at least 5 decoding iterations, as seen in figure 4
- The system shall use the "Belief Propagation" algorithm
- The system shall perform at a code rate of ½
- The system codeword size for FPGA implementation shall be 1008 bits long
- The system message word size for FPGA implementation shall be 504 bits long
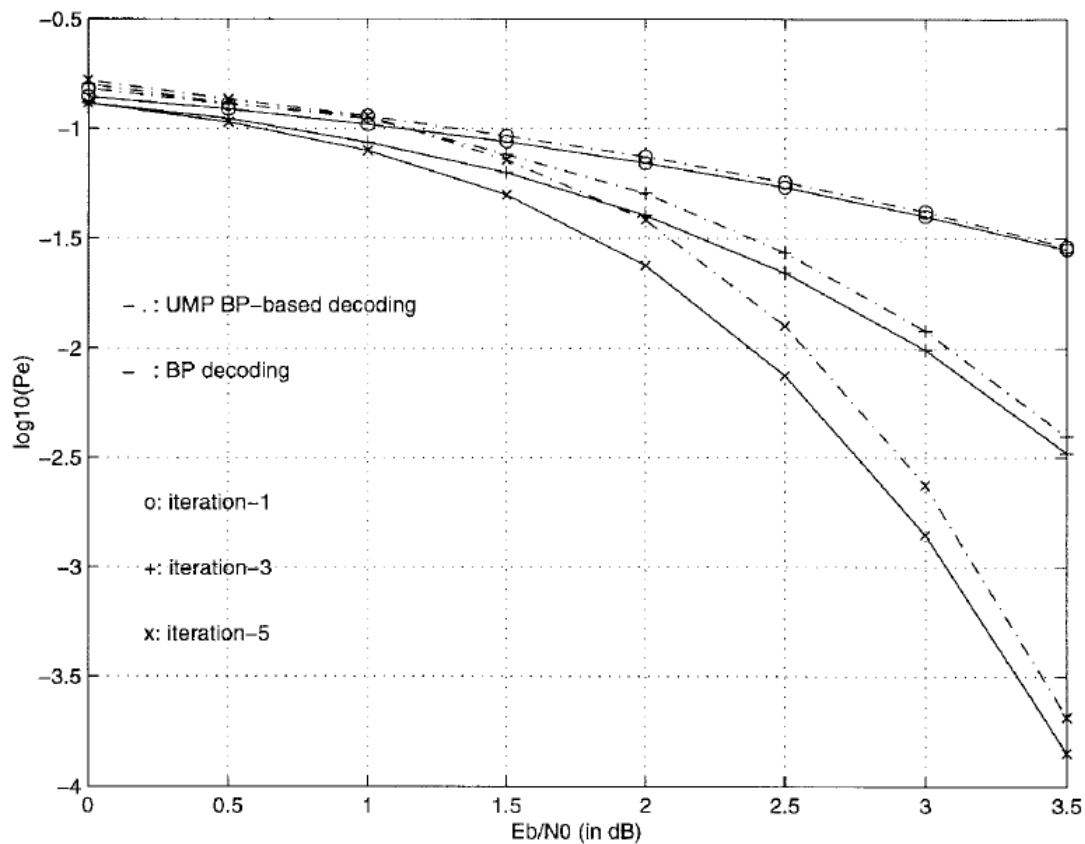- The system shall use less than 500,000 logic gates on the FPGA

Figure 4.   Error Performance vs. Signal/Noise Ratio [3]

References

[1] Leiner, Bernard M.J, "*LDPC Codes – a Brief Tutorial*". University of Arizona, April 2005.

[2] Sklar, Bernard, "Digital Communications: Fundamentals and Applications", 2nd Edition, Prentice Hall, 2000.

[3] Fossorier, Marc P. C., Miodrag Mihaljevi´c, and Hideki Imai. *Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation. EE296M - Error Correcting Coding*. San Jose State University, 2005. Web. 25 Oct. 2012. <http://www.engr.sjsu.edu/rmorelos/ee296f05/Fossorier1999.pdf>.