Hard decision decoding of binary low-density codes based on extended Hamming codes

Alexey I. Kuznetsov
Institute for Information Transmission Problems
Russian Academy of Sciences

Introduction

In this paper we investigate a possibility to use a hard decision low-density decoder for high-bandwidth optical channels.

The code in question is a binary code similar to Gallager's classic LDPC code, but parity-checks are replaced with extended Hamming codewords.

Contents

Code definition

H_1 – parity-check matrix of an extended Hamming code (length n_0)

H_m – matrix containing H_1 m times (as specified below)

P_i – permutation matrixes

$$H_m = \begin{bmatrix} H_1 & 0 & 0 & \dots & 0 \\ 0 & H_1 & 0 & \dots & 0 \\ \dots & & & & \\ 0 & 0 & 0 & \dots & H_1 \end{bmatrix} \qquad H = \begin{bmatrix} H_m P_1 \\ H_m P_2 \\ \dots \\ H_m P_l \end{bmatrix}$$

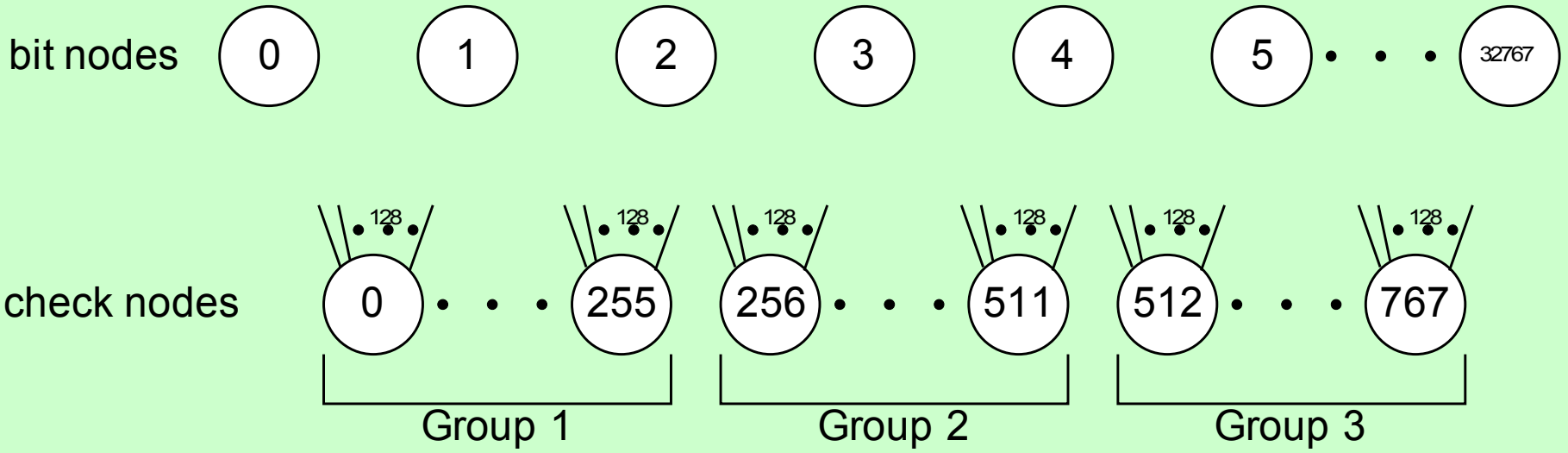
We shall restrict ourselves to the following code parameters:

$n_0 = 128$

$m = 256$

$l = 3$

Graph



The check nodes are divided into three groups. Each check node represents an extended Hamming codeword.

Each bit node is connected with one and only one check node in each group.

Decoding

For each bit, we decode all the three component codewords containing this bit. The result for each codeword can be:

- 0 – syndrome is zero
- 1 – syndrome isn't zero and this particular bit is marked as erroneous
- F – the component decoder failed to decode this codeword
- E – the component decoder found a error in other position

Actions:

- There is one 1 and no other is 0 => correct
- There is two 1's => correct
- Otherwise => don't correct

The action is repeated for every bit until the decoder can't correct any more errors..

Numeric simulation

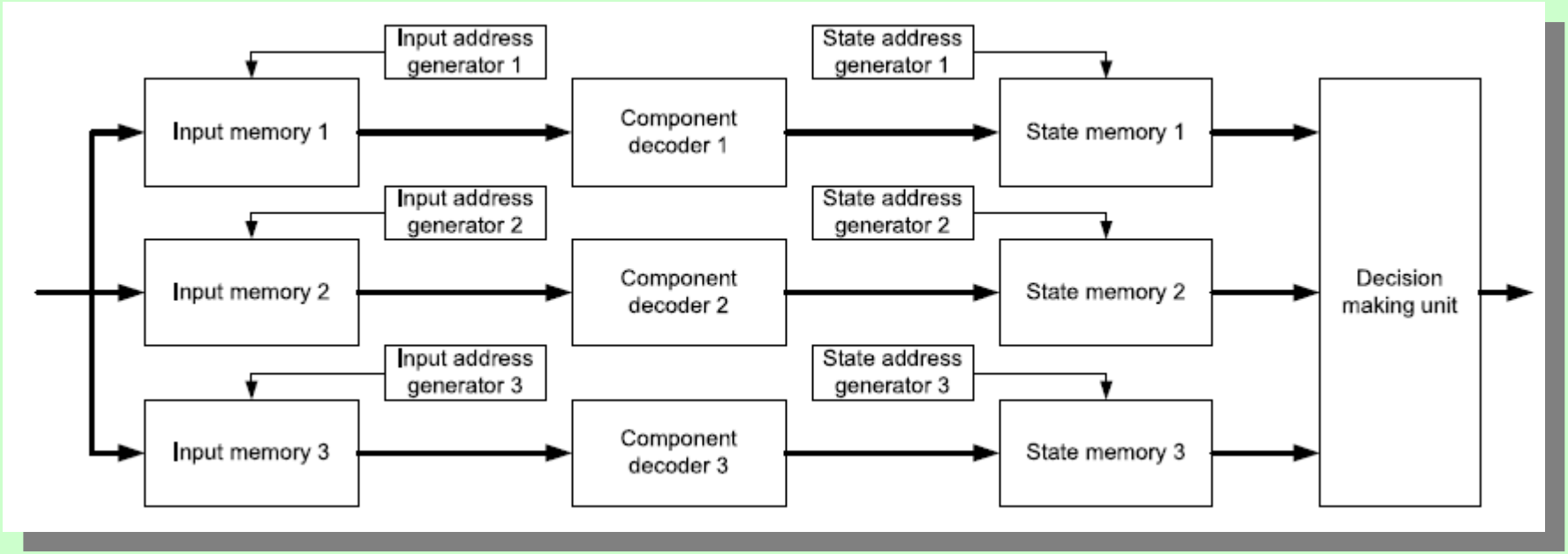
The following bit error probabilities were obtained during simulation.

Two cases were simulated:

- Case 1 – All permutations are random
- Case 2 – Permutation 2 is chosen arbitrarily

p_{in}	p_{out} (case 1)	p_{out} (case 2)
0.011	6.9e-6	4.6e-6
0.010	8.2e-7	3.2e-7
0.009	4.3e-7	9.8e-8

Hardware implementation: Architecture



A simple sequential approach can be employed to decode this code in hardware (either FPGA or ASIC).
Each component code decoder processes one bit per cycle.

Hardware implementation: Resource utilization

The amount of logic cells utilized by this design is defined mostly by memory requirements, since the other logic uses a few cells.

The maximum memory requirement is about 3,5 Mbit, which is not very much even for modern FPGAs. (for example, Xilinx Virtex-4 chips contains up to 10 Mbit of block RAM).

Sequential design: pro et contra

Advantages:

- Simplicity
- Humble hardware requirements

Disadvantages:

- A maximum of 1 bit per cycle can be processed, so the processing speed is limited by the frequency of semiconductor devices, making it virtually impossible to reach processing speeds above a few Gbit/s.

A problem

In order to overcome the processing speed limitation on sequential design, one should employ parallelism. It would be desirable, for example, to have dedicated component decoder for every component code.

Unfortunately, the task is challeging since it uses a lot of data transfers. Such algorithms can not be easily paralleled.

Conclusion

In this work a method was presented to decode a low-density code based on extended Hamming component codes in hardware. It was shown that this method has inherent limitations which hinders applications of this code on high-speed optical links. These limitations can not be easily overcomed.

It was also stated that the possible way to solve this problem is to find a reasonable parallel implementation of this algorithm.