Introduction

A low-density parity-check (LDPC) code is a linear error correcting code, a method of transmitting a message over a noisy transmission channel by adding redundancy to the data bits. LDPC codes are capacity approaching codes, which means that practical constructions exist that allow for low error probability decoding at noise close to the theoretical maximum (the Shannon limit). LDPC codes are finding increasing use in data-corrupting noise is desired (10 Gbps Ethernet, DVBS-S, etc).

Gallager invented “regular” LDPC codes in 1960 in his Ph.D. thesis. The codes are often represented using a bipartite graph. An example is shown here. The codes are also called \((c,d)\)-regular codes if the bit nodes have degree \(c\) and the check nodes have degree \(d\).

The goal of this project is to implement a simple LDPC decoder for a \((3,4)\)-regular LDPC code using a message-passing decoding technique called the “Gallager A decoding algorithm”.

On the left is shown the bi-partite graph of an LDPC decoder. With no channel errors, all the parity-check equations (shown at the right) will all be satisfied. The information to be sent is represented by bits \(B_1, B_2\), and the redundant bits (parity bits), \(P_1, P_2, \ldots, P_6\). The rate of this code is therefore \(\frac{1}{4}\).
The Decoding Process

The decoding proceeds in iterations. Each iteration has two steps. In the first step of each iteration, all the bit nodes send messages to the check nodes. In the second step, all the check nodes send messages back to bit nodes. More precisely, in Gallager A algorithm,

the bit node sends to all connected check nodes its estimate of its own value $B$, which is

- The received bit $Y_i$ from the channel in the first iteration.
- The following function in any iteration after the first iteration

$$f_{\text{bit-to-check}}(B, C_1, \ldots, C_c) = \begin{cases} C_1 & \text{if } C_1 = C_2 = \ldots = C_c \\ B & \text{otherwise} \end{cases}$$

where $C_1, \ldots, C_c$ are the messages from the check nodes connected to the bit node.

The check node sends to all connected bit nodes its estimate of the value of the respective bit node, which is

$$f_{\text{check-to-bit}}(B_{j1}, B_{j2}, \ldots, B_{jd}) = B_{j1} \oplus B_{j2} \oplus \ldots \oplus B_{jd}$$

where the XOR is over all the input bits (received from the bit nodes in the current iteration) except the bit node to which the message is being sent. It is important to note here that all the bit nodes connected to a check node receive a different function of the inputs to a check node.

The decoding terminates when the messages stabilize (further iterations do not change the messages), or if the number of iterations reaches a pre-specified maximum.
From Algorithm to Circuit Implementation

From the previous introduction, a LDPC decoder basically consists of check nodes, bit nodes and wires. In Fig. 1 and Fig. 2, you can find the high-level block diagrams of the check node and bit node of (3, 4)-regular codes. The operation performed inside these nodes can be mapped to circuit level implementation. In this project, you are going to learn how to implement a full LDPC decoder from algorithm to digital circuits.

**Check Node**

*Check Node: Receive 4 bits from 4 Bit Nodes and send back parity check results*

*Major Operation: XOR, simple control logics*

**Bit Node**

*Bit Node: Receive from 3 Check Nodes and channel and send the result back to those Check Nodes*

*Major Operation: Bit-to-Check function, update value, simple control logics*
Project Overview

The first task is to find your partners – the project will be executed in **groups of 3 students**. Smaller and larger groups (each of which is discouraged) need special permission from Prof. Rabaey.

The project will be executed in three phases.

- Phase 1: Design of Check Node (March 5 ~ March 15)
- Phase 2: Design of Bit Node and Wire Routing (March 15 ~ April 9)
- Phase 3: Design of a Full LDPC Decoder with Wires (April 9 ~ May 5)

For Phases 1 and 2, the outcome will be a simple report, while Phase 3 culminates in the creation of a poster and a short interview, where you will have the opportunity to describe how brilliant your design is.

Description of Project Phase 1

In this phase, you will follow the principles of LDPC decoding and work on a step-by-step exercise of decoding iterations. After finishing the exercise, you will be given the opportunity to design the check node of LDPC decoder in Fig. 1 by using any static logic family. **The goal is to minimize the propagation delay of the check node.**

More specifically, you are asked to perform following tasks:

1. Do an exercise on LDPC decoding (given in the report template). Walk through the decoding process step by step and find the decoding results after iterations.
2. Design the check node using any static logic family.
3. Identify the critical path(s) in your design.
4. Size the logic gates in your design to minimize propagation delay of the critical path(s) in check node.
5. Simulate your design in Cadence SPECTRE and report the critical path propagation delay.

No layout work is required in Phase 1.

Design Constraints

- **Technology:** GPDK 90nm in the EE141 Lab Library;
- **Supply Voltage:** ≤ 1.2V;
- **Input Capacitance:** Loading on each check node’s input: ≤ 4C, where C is the gate capacitance of a unit-size inverter (NMOS: 120nm/100nm, PMOS: 240nm/100nm);
- **Load Capacitance:** Loading on each check node’s output comes from the input capacitance of the bit node and the wire loading between check nodes and bit nodes.
  1. The input capacitance of the bit node is 4C, same as the check node.
  2. The wire loading can be estimated by hand calculation. Assume the wire is 0.25 mm long with minimum width using Metal 3 layer. The wire capacitance table can be found on the project webpage.
• **Noise Margins:** The noise margins should be at least 10% of the supply voltage.
• **Rise and Fall Times:** All input signals from ideal signal sources in simulation have rise and fall times of 100 ps.

**Report**

Please use the report template provided at the website. Be sure to justify important design decisions and emphasize all the vital information. **Organization, conciseness, and completeness are of paramount importance.** Good reports are short and to the point. Make sure to include and annotate important plots to illustrate your effort. Do not repeat the information we already know. Make sure to fill out the cover-page and use the correct units.

**Grading**

The quality of the report is a major part of the grade.

For Phase 1, the grade will be equally divided over the correctness and completeness of the results, and the quality of the report.

**Have fun, and good luck!**