Structured LDPC Codes Design and Its Reconfigurable Decoder Implementation

Min Li (PhD Advisor: Aylin Yener, WCAN, Electrical Engineering)

Problem Statement and Motivation

➢Random LDPC codes are widely studied for the past few years, which are shown to have powerful error-correcting capability, i.e., only 0.0045 dB away from Shannon limit at rate ½;

➢Structured Irregular-Repeat-Accumulate (SIRA) LDPC codes have lower implementation complexity, which makes them more attractive for applications in practice than random LDPC codes;

➢Two problems are solved here:
- ✓To construct multi-rate SIRA (***MR-SIRA***) LDPC codes with hardware-constraint;
- ✓To design an efficient ***reconfigurable decoder*** for multi-rate block-structure LDPC codes.

Key Tools and Rules

➢Code Construction
- ✓Node Degree Distribution Optimization: Gaussian Approximation (GA) with Constraint;
- ✓Binary Mother Matrix Optimization: Progressive Edge Generation (PEG);
- ✓Objective Block Matrix Optimization: Optimal Cycle Distribution Rule

➢Reconfigurable Decoder Design
- ✓Programmable processing modules
- ✓Programmable logic control modules
- ✓Minimum memory usage rule
- ✓Minimum logic gates rule
- ✓Scalable

SIRA LDPC Codes

➢SIRA Codes
- ✓Simple structure
  - ◆Composed of small blocks of circulant permutation matrix
  - ◆Own double-diagonal property
- ✓Ensure fast encoding through accumulator
- ✓Allow low-complexity decoding scheme in practice

Question:

How to fill in shifting index p_{ij} to obtain an H matrix with the optimal performance ?

$$\mathbf{E_H} = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,N-M} & 0 & \ddots & -1 & -1 \\ p_{1,0} & p_{1,1} & \cdots & \vdots & 0 & 0 & \ddots & -1 \\ \vdots & \vdots & \ddots & p_{i,N-M} & \ddots & 0 & 0 & -1 \\ p_{M-2,0} & p_{M-2,1} & \cdots & \vdots & -1 & \ddots & 0 & 0 \\ p_{M-1,0} & p_{M-1,1} & \cdots & p_{M-1,N-M} & -1 & -1 & \ddots & 0 \end{bmatrix}_{M\times N}$$

$M\times(N-M)$ → $\mathbf{H_s}$; $M\times M$ → $\mathbf{H_p}=[\mathbf{H_{pa}}\ \mathbf{H_{pb}}]$

Code Construction Algorithm

Key Steps

Step 1. Given the objective code parameter such as rate, find the optimal degree distribution using Gaussian Approximation with hardware-constraint;

Step 2. Given the node degree distribution, construct a binary mother matrix M($\mathbf{H}_0$) to fix the non-zero positions in the matrix. Apply the improved PEG concept during this process in order to obtain an optimal mother matrix;

Step 3. Construction of the objective matrix E(**H**) based on M($\mathbf{H}_0$): Fill in elements for each non-zero position selected from {0, 1, …, L-1} and survive the candidate that maximizes the girth;

Step 4. Given E(**H**), do cycle distribution analysis. Repeat *Step 2* and *Step 3* until the optimal objective matrix is found in a given number of trials.

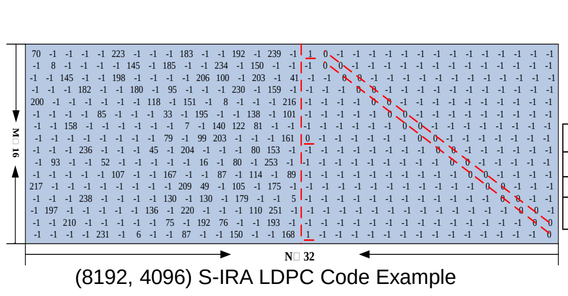
An Example

➢Code Parameters
- ➢R= ½, M=16, N=32, L= 256;
- ➢Node degree distribution:

V node: $\lambda(x) = 0.2655x + 0.2389x^2 + 0.4956x^6$

C node: $\rho(x) = 0.9292x^6 + 0.0708x^7$

➢Construction Result:



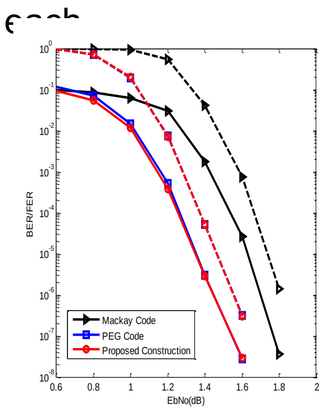
(8192, 4096) S-IRA LDPC Code Example

Cycle Distribution Comparison

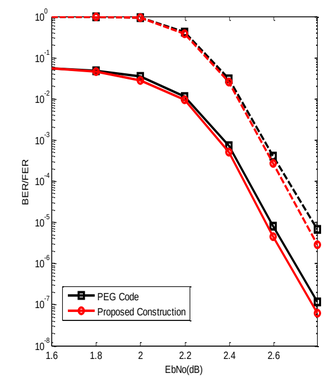
Cycle Dis. (%)	6	8	10	12
Mackay	6.53	42.46	50.91	0.1
PEG	0	77.97	22.03	0
Our Code	0	53.13	46.87	0

Performance of Our Codes

➢Simulation: BP decoding, 50 iterations for each



Rate = 1/2 with length 8192

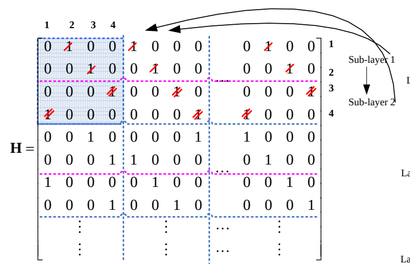


Rate = 3/4 with length 8192

➢Note: MR-SIRA codes show better error-correcting performance than Mackay codes and are at least no worse than PEG codes. But both encoding and decoding of MR-SIRA codes are much simpler due to their inherent block structure.

Efficient Decoding for Our Codes

➢ Layered-Shuffle BP-Based Decoding

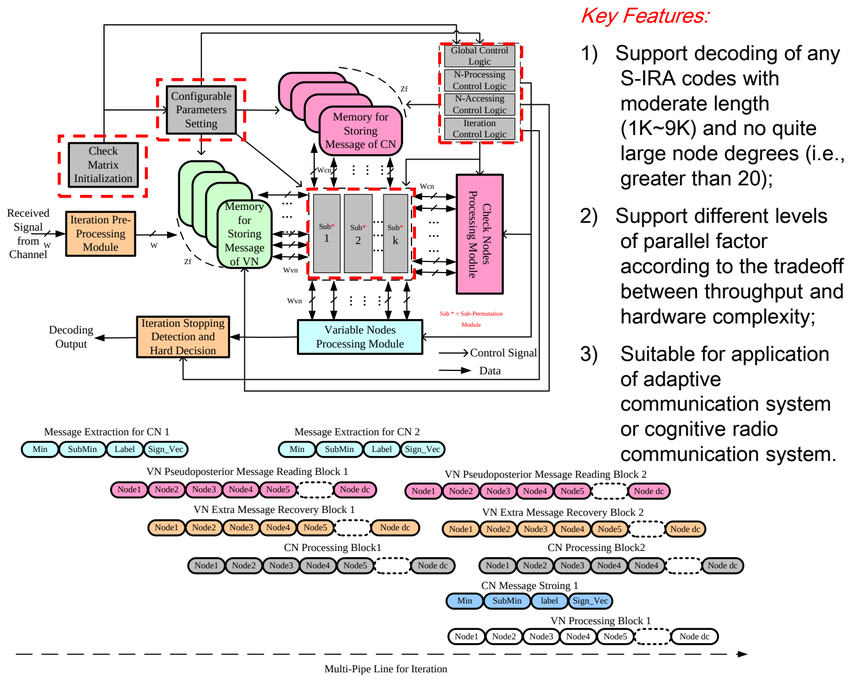


Algorithm Description

Step 1 Initialize $LLR_{ini,j}$
Step 2 Iteration :
  While (iter < MAXITER) & (success_flag!=1)
    for m = 1: M
      horizontal process:
      for l = 1: L_{layer}
        for k = 1: L/L_{layer}
          for each j belongs to $N((m-1)L + (l-1)L_{layer} + k)$
            $LQ_{k,j} = LLR_{post,(iter-1),k,j} - LR_{(iter-1),k,j}$
          end
          for each j belongs to $N((m-1)L + (l-1)L_{layer} + k)$
            update $LR_{iter,k,j} = f(LQ_{k,j})$
          end
          for each j belongs to $N((m-1)L + (l-1)L_{layer} + k)$
            update $LLR_{post,\ iter,k,j} = f(LR_{iter,k,j})$
          end
        end
      end
    end
  make check sum calculation
  if (check_sum_valid)
    success_flag = 1
  else
    success_flag = 0
    iter++
  end
  end
Step 3 Output final decoding bit streaming

- Sub-layers (layers) are scheduled sequentially;
- Once each sub-layer finishes updating, the variable nodes connected with this layer are updated immediately so that the latest information from check nodes is effectively propagated.

Reconfigurable Decoder Design



Key Features:

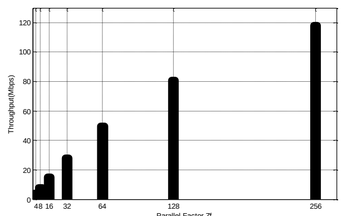
1) Support decoding of any S-IRA codes with moderate length (1K~9K) and no quite large node degrees (i.e., greater than 20);
2) Support different levels of parallel factor according to the tradeoff between throughput and hardware complexity;
3) Suitable for application of adaptive communication system or cognitive radio communication system.

Throughput Evaluation

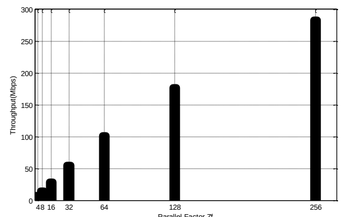
➢Notation Table:

R	Code rate	T_{idle}	Number of idle clocks in the decoding pipeline
M	Number of code row blocks	$ITER_{max}$	Number of maximum iterations
N	Number of code column blocks	f_{max}	Maximum system frequency
M_{sub}	Number of code sub layers	T_{sub}	Number of clocks for each sub layer's iteration

$$Throughput = \frac{f_{\max} \times N \times L \times R}{\left[M \times M_{sub} \times T_{sub} + (M-1) \times T_{idle}\right] \times ITER_{\max}} \text{ Mbps}$$



Rate 1/2 with length 8192, f_{max} = 120 M



Rate 3/4 with length 8192, f_{max} = 120 M

Conclusion and Potential Future Work

➢ Conclusion:

✓The proposed MR-SIRA codes and reconfigurable decoder design are attractive for applications that require both excellent code performance and low implementation complexity;

➢ Possible Work: (1) Structured codes and implementation issues of decoding for LDPC on Gf (q); (2) Application of BP to compressive sensing; (3) Extended application of LDPC codes, i.e., cooperative network coding scenario.

This work was my previous research which had been done in early 2008 before I joined in WCAN.