Two-Level Code Construction Using LDPC Codes

Osso Vahabzadeh
TexasLDPC Inc.
Overview

• Encoding
• Decoding
• Memory Requirements
• Latency
• Simulation Results
• Summary
Encoding

\[ u_1 = [u_{1,1}, u_{1,2}, \ldots, u_{1,k_1}] \quad \Rightarrow \quad c_1 = [c_{1,1}, c_{1,2}, \ldots, c_{1,n_1}] \]

\[ u_2 = [u_{2,1}, u_{2,2}, \ldots, u_{2,k_2}] \quad \Rightarrow \quad c_2 = [c_{2,1}, c_{2,2}, \ldots, c_{2,n_2}] \]

\[ \vdots \]

\[ u_T = [u_{T,1}, u_{T,2}, \ldots, u_{T,k_T}] \quad \Rightarrow \quad c_T = [c_{T,1}, c_{T,2}, \ldots, c_{T,n_T}] \]

\( T \) is the page size
Encoding

\[ u_1 = [u_{1,1}, u_{1,2}, \ldots, u_{1,k_1}] \rightarrow c_1 = [c_{1,1}, c_{1,2}, \ldots, c_{1,k_1}] \rightarrow \text{Interleaver } \pi_1 \rightarrow c_{1,1}^{(\pi_1)}, c_{1,2}^{(\pi_1)}, \ldots, c_{1,k_1}^{(\pi_1)} \]

\[ u_2 = [u_{2,1}, u_{2,2}, \ldots, u_{2,k_2}] \rightarrow c_2 = [c_{2,1}, c_{2,2}, \ldots, c_{2,k_2}] \rightarrow \text{Interleaver } \pi_2 \rightarrow c_{2,1}^{(\pi_2)}, c_{2,2}^{(\pi_2)}, \ldots, c_{2,k_2}^{(\pi_2)} \]

\[ \vdots \]

\[ u_T = [u_{T,1}, u_{T,2}, \ldots, u_{T,k_T}] \rightarrow c_T = [c_{T,1}, c_{T,2}, \ldots, c_{T,k_T}] \rightarrow \text{Interleaver } \pi_T \rightarrow c_{T,1}^{(\pi_T)}, c_{T,2}^{(\pi_T)}, \ldots, c_{T,k_T}^{(\pi_T)} \]

\( T \) is the page size
Encoding

For simplicity, one encoder circuitry is used

Only $p_{aux}$ is stored

$T$ is the page size
The Code Graph

Edge Interleaver

\[ c_{1,1}, c_{1,2}, \ldots, c_{1,n_1} \]

Edge Interleaver

\[ c_{2,1}, c_{2,2}, \ldots, c_{2,n_2} \]

Edge Interleaver

\[ c_{T,1}, c_{T,2}, \ldots, c_{T,n_T} \]
The Code Graph
The Code Graph

Edge Interleaver

Interleaver ($\pi_1$)

$c_{1,1}$ $c_{1,2}$ $\ldots$ $c_{1,n_1}$

Subset of $n_1$

Edge Interleaver

Interleaver ($\pi_2$)

$c_{2,1}$ $c_{2,2}$ $\ldots$ $c_{2,n_2}$

Subset of $n_2$

Edge Interleaver

Interleaver ($\pi_T$)

$c_{T,1}$ $c_{T,2}$ $\ldots$ $c_{T,n_T}$

Subset of $n_T$

Additional Overhead

$c_{a,1}$ $c_{a,2}$ $c_{a,p_a}$
Step 1. Decode the $T$ primary codewords in one memory page $d_f$: number of decoding failures

- If $d_f = 0$,
  decode the $T$ primary codewords in the next page;
- else
  for $m = 1, \ldots, d_f$
    \textbf{Step 2.} If the auxiliary codeword has not been decoded yet, compute its LLRs and decode it.
    
    \textbf{Step 3.} Update the LLRs of the failed primary codeword based on the results of Step 2 and decode it one more time.
Memory Requirements: One-Level Code

- Double buffered Q memory
- Double buffered HD memory
- Qsign memory
- FS memory

The block serial layered decoder
(the most memory efficient decoder architecture)
Memory Requirements: Two-Level Code

- Only one decoder circuit used
- Double buffered Q memory
- Double buffered HD memory
- Qsign memory
- FS memory
- HD memory for the failed codeword
- ROM to store the connectivity data of the auxiliary and primary codes
Latency

- Latency introduced only when the auxiliary codeword utilized
- Sources of latency:
  - Load the HDs related to the last primary codeword in the auxiliary codeword’s HD memory ($bc$ : block columns)
  - Read the parity bits of the auxiliary codeword ($p$ : parity blocks)
  - Decode the auxiliary codeword (decoding algorithm, parallelization)
  - Load the HD detection results of the failed codeword for decoding ($bc$)
  - Decode the failed codeword one more time (decoding algorithm, parallelization)
Example

• FER = $10^{-4}$ : 1 out of 10000 packets fails on the average
• Rate = 0.91, block columns = 66, parity blocks = 6, non-zero circulants = 264
• Parallelization = circulant size, Page size = 16, Average iterations = 3
  o Load the HDs of the last primary codeword: 66 clks
  o Read the parity bits of the auxiliary codeword: 6 clks
  o Decode the auxiliary codeword: $3 \times 264 = 792$ clks
  o Load the HD detection results of the failed codeword: 66 clks
  o Decode the failed codeword one more time: 792 clks
  o Overall page latency = $1722 / (16 \times 792) = 13.6\%$ (for 0.01% of the packets)
  o For 99.99% of the packets, the latency same as the one-level system
Simulation Results (1)

- Rate = 0.83
- Length = 1 KB
- $T = 16$
- Circulant size = 140
- Binary LDPC codes
Simulation Results (2)

- Rate $= 0.91$
- Length $= 1$ KB
- $T = 16$
- Circulant size $= 140$
- Binary LDPC codes
Summary

• A two-level code construction scheme based on LDPC codes presented
• Improved frame error rate (FER) performance compared to one-level scheme
• Small memory requirements, particularly, when utilized with memory efficient architectures
• Negligible latency at low FER regime
Thank You!