Abstract: We study decoding algorithms of low-density parity-check (LDPC) codes defined over GF(q). Logarithm-domain sum-product (Log-SP) algorithms were proposed for reducing quantization effects of SP algorithm in conjunction with FFT. Since FFT is not applicable in the logarithm domain, the computations required at check nodes in the Log-SP algorithms are computationally intensive. What is worse, check nodes usually have higher degree than variable nodes. As a result, most of the time for decoding is used for check node computations, which leads to a bottleneck effect. In this paper, we propose a Log-SP algorithm in the Fourier domain. With this algorithm, the roles of variable nodes and check nodes are switched. The intensive computations are spread over lower-degree variable nodes, which can be efficiently calculated in parallel.

Non-binary LDPC codes

- Non-binary LDPC codes are defined over Galois field GF(2^m). Let \( x \in GF(2^m) \) and \( Hx = 0 \).
- \( H = (h_{e,v}) : M \times N \) sparse matrix over GF(2^m).
- Efficiently decodable in parallel by the iterative decoding algorithm on the Tanner graph.

Notation

\[ [p_{1} \otimes p_{2}](x) := \sum_{\xi \in GF(q)} p_{1}(\xi)p_{2}(x - \xi) \]
\[ [\lambda_{1} \otimes \lambda_{2}](x) := \ln(\sum_{\xi \in GF(q)} e^{\lambda_{1}(\xi)+\lambda_{2}(x-\xi)}) \]
\( V_{c} := \{ v \in [1, M] \mid h_{e,v} \neq 0 \} \)
\( C_{v} := \{ c \in [1, M] \mid h_{v,c} \neq 0 \} \)
\( \tilde{\rho}_{e,v}(x) := p(h_{e,v}x) \)
\( \tilde{\lambda}_{e,v}(x) := p(h_{e,v}x) \)
\( \tilde{\rho}_{v,c}(x) := p(h_{v,c}x) \)
\( \tilde{\lambda}_{v,c}(x) := p(h_{v,c}x) \)

Difficult of Decoding High-rate LDPC codes

- High rate codes have higher-degree \( \square \) nodes.
- Complex computations are concentrated in high-degree nodes, which gives rise to bottleneck effect.

Problems

- \( \square \) computations are much complex than \( \bigcirc \) computations.
- High rate codes have higher-degree \( \square \) nodes.
- Complex computations are concentrated in high-degree nodes, which gives rise to bottleneck effect.

Multiplications are computationally expensive and need high precision numbers.

New Algorithm Proposal

<table>
<thead>
<tr>
<th></th>
<th>( \bigcirc )</th>
<th>( \square )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Product</td>
<td>( \times )</td>
<td>( \square )</td>
</tr>
<tr>
<td>Log-domain SP</td>
<td>( \bigcirc )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( F )-domain SP\textsuperscript{NEW}</td>
<td>( \times )</td>
<td>( \bigcirc )</td>
</tr>
</tbody>
</table>

\( F \)-domain and Log-\( F \)-domain algorithm is proposed.
- \( \square \) computations and \( \bigcirc \) computations are switched.
- Complex computations are performed by \( \bigcirc \) which have less incoming edges.
- The bottleneck effect is relieved!!.

Iterate:

- Initialize:
  \[ \tilde{\rho}_{e,v}^{(0)}(x) := p(h_{e,v}x) \]
  \[ \tilde{\lambda}_{e,v}^{(0)}(x) := p(h_{e,v}x) \]

- Iterate: \( \tilde{\rho}_{e,v}^{(l+1)}(x) = \prod_{\nu \in V_{c} \setminus \{v\}} \tilde{\rho}_{\nu}^{(l)}(x) \)
  \[ \tilde{\lambda}_{e,v}^{(l+1)}(x) = -\eta + \lambda_{e,v}^{(l)}(x) + \sum_{c \in C_{v} \setminus \{e\}} \lambda_{v,c}^{(l)}(x) \]

- Initialize:
  \[ \hat{\rho}_{e,v}^{(0)}(x) := p(h_{e,v}x) \]
  \[ \hat{\lambda}_{e,v}^{(0)}(x) := p(h_{e,v}x) \]

- Iterate: \( \hat{\rho}_{e,v}^{(l+1)}(z) = \prod_{\nu \in V_{c} \setminus \{v\}} \hat{\rho}_{\nu}^{(l)}(z) \)
  \[ \hat{\lambda}_{e,v}^{(l+1)}(z) = -\eta + \lambda_{e,v}^{(l)}(z) + \sum_{c \in C_{v} \setminus \{e\}} \lambda_{v,c}^{(l)}(z) \]

- Initialize:
  \[ \hat{\mu}_{e,v}^{(0)}(z) := \ln |\hat{\lambda}_{e,v}^{(l)}(z)| \]
  \[ \hat{\sigma}_{e,v}^{(0)}(z) := \text{sgn}(\hat{\lambda}_{e,v}^{(l)}(z)) \]

- Iterate: \( \hat{\mu}_{e,v}^{(l+1)}(z) = -\eta + \sum_{\nu \in V_{c} \setminus \{v\}} \hat{\mu}_{\nu}^{(l)}(z) \)
  \[ \hat{\sigma}_{e,v}^{(l+1)}(z) = \prod_{\nu \in V_{c} \setminus \{v\}} \hat{\sigma}_{\nu}^{(l)}(z) \]
  \[ \nu_{\pm}(z) = \ln \sum_{z' \in Z_{\pm}} \exp(\hat{\mu}_{e,v}^{(l)}(z') + \hat{\mu}_{e,v}^{(0)}(z + z')) \]
  \[ Z_{\pm} = \{ z' \in \{0,1\}^m | \pm \hat{\sigma}_{e,v}^{(0)}(z') \hat{\sigma}_{e,v}^{(0)}(z + z') > 0 \} \]