OPTIMIZING ITERATIVE DECODING OF LOW-DENSITY PARITY CHECK CODES ON PROGRAMMABLE PIPELINED PARALLEL ARCHITECTURES

- INTRODUCTION
- ITERATIVE LDPC DECODING
- THE PROPOSED ARCHITECTURE
- THE PROPOSED SOLUTION
LDPC code is defined by a sparse MxN parity check matrix $H$. 

$$
\begin{bmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
$$

LDPC code can be represented by a bipartite graph $G=(V,F,E)$. 
INTRODUCTION

There is an edge $e_{i,j} \in E$ between $v_i$ and $f_j$ corresponding to non-zero entries. It consists of $N$ variable nodes $v_i \in V$ corresponding to columns and $M$ function nodes $f_j \in F$ corresponding to rows.
ITERATIVE LDPC DECODING

- Sum product (message passing) algorithm
- Updates of all function nodes are done in the first phase
- Updates of all variable nodes are done in the second phase

\[
\text{LLR}(\mu_{f_j \to v_i}) = \text{LLR}(r_{ij}) = \log \left( \frac{r_{ij}(1)}{r_{ij}(0)} \right)
\]

This is the message from function node \( f_j \) to variable node \( v_i \)
ITERATIVE LDPC DECODING

$$\text{LLR}(\mu_{v_i \rightarrow f_j}) = \text{LLR}(q_{ij}) = \log \left( \frac{q_{ij}(1)}{q_{ij}(0)} \right)$$

This is the message from variable node $v_i$ to function node $f_j$

The algorithm consists of the following steps:

**Initialize:**

$$\text{LLR}_{\text{prior}}(v_i) = \log \left( \frac{p_{i_{\text{prior}}}(1)}{p_{i_{\text{prior}}}(0)} \right)$$
ITERATIVE LDPC DECODING

 Phase 1 of iteration: Variable node processing

\[ LLR\left(q_{ij}\right) = \sum_{f_j \in Nbr(v_i) \setminus \{f_j\}} LLR\left(r_{ij}\right) + LLR\; prior\left(v_i\right) \]
ITERATIVE LDPC DECODING

Phase 2 of iteration: Function node processing

\[ LLR(r_{ij}) = \phi^{-1} \left( \sum_{v_i \in Nbr(f_j) \setminus \{v_i\}} \phi \left( LLR(q_{i,j}) \right) \right) \]

\[ \prod_{v_i \in Nbr(f_j) \setminus \{v_i\}} \text{sgn}(LLR(q_{i,j}))(−1)^{|Nbr(f_j)|} \]

where

\[ \phi(x) = \phi^{-1}(x) = -\log \left( \tanh \left( \frac{1}{2} x \right) \right) = \log \frac{\exp(x) + 1}{\exp(x) - 1} \]
THE PROPOSED ARCHITECTURE

- An example
- The system
- Mathematical model
n fun2var memory banks

n variable processors

n var2fun memory banks

n function processors
An Example

\[ H = \begin{pmatrix}
  11011 \\
  01100 \\
  00011 \\
\end{pmatrix} \]
An Example
An Example
An Example

1 2 3 4 5
A B C

4,1 2,3 5
4-A 2-A 5-A

A B C
An Example
An Example
An Example

t=1  4-A  3-B  5-C

1-A  4-C  5-C

3-B  2-B  5-A
An Example

1  2  3  4  5
A  B  C

\begin{align*}
t=1 & \quad 4-A \quad 3-B \quad 5-C \\
t=2 & \quad 2-A \quad - \quad 4-C
\end{align*}
An Example

\[
\begin{align*}
    t=1 & & 4-A & & 3-B & & 5-C \\
    t=2 & & 2-A & & - & & 4-C \\
    t=3 & & 5-A & & - & & - \\
\end{align*}
\]
An Example

<table>
<thead>
<tr>
<th>t=1</th>
<th>4-A</th>
<th>3-B</th>
<th>5-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=2</td>
<td>2-A</td>
<td>-</td>
<td>4-C</td>
</tr>
<tr>
<td>t=3</td>
<td>5-A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>t=4</td>
<td>1-A</td>
<td>2-B</td>
<td>-</td>
</tr>
</tbody>
</table>

A B C

4,1 2,3 5

1-A 3-B 5-C
4-C 2-B 2-A
4-A

A B C
The system

Constraints:
- Every node is processed by a single processor and only once.

\[ X_{v,i,j,k} = 1 \text{ if variable node } i \text{ is the } k\text{th node processed by variable processor } j, \]
\[ X_{v,i,j,k} = 0 \quad \text{otherwise} \]

\[ X_{f,i,j,k} = 1 \text{ if function node } i \text{ is the } k\text{th node processed by function processor } j, \]
\[ X_{f,i,j,k} = 0 \quad \text{otherwise} \]

\[ \forall i \quad \sum_{j=0}^{n-1} \sum_{k=0}^{N-1} X_{v,i,j,k} = 1 \]
\[ \forall j \quad \sum_{j=0}^{n-1} \sum_{k=0}^{M-1} X_{f,i,j,k} = 1 \]
Mathematical model

Given the H matrix solve the combinatorial optimization problem of minimizing minimizing latency of a single decoding iteration
Mathematical model

- Deciding which function nodes are assigned to each function processor (assignment problem)
- Deciding in which order these nodes are processed (scheduling problem)
Mathematical model

- Deciding how the n processors read from the n memory banks each cycle.
Mathematical model

\[
L_{vp_j} = \text{IPD } v + \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} \deg V \left( i \cdot X_{v_i,j,k} \right) + \sum_{t=0}^{L_{vpj}} Z_{v_{j,t}} + \]

\[
\sum_{k=0}^{N} \sum_{i=0}^{N-1} \max \left( 0, \deg V \left( i \cdot X_{v_i,j,k} \right) - \deg V \left( i \cdot X_{v_i,j,k} \right) \right)
\]

\[Z_{v_{j,t}} = 1 \text{ if variable processor } j \text{ stalls at cycle } t,\]
\[Z_{v_{j,t}} = 0 \text{ otherwise}\]
Mathematical model

\[ L_{fp_j} = IPD f + \sum_{i=0}^{M-1} \sum_{k=0}^{M-1} \deg F(i.Xf_{i,j,k}) + \sum_{t=0}^{L_{fp_j}} Z_v + \]

\[ \sum_{k=1}^{M} \sum_{i=0}^{N-1} \max(0, \deg F(i.Xf_{i,j,k}) - \deg F(i.Xf_{i,j,k})) \]

Min \( X_v, X_f \) \( \{ L_{total} = \max_j (L_{vp_j}) + \max_j (L_{fp_j}) \} \)
The Proposed Solution

- Solving The Mapping Problem
- Solving The Scheduling Problem
Solving The Mapping Problem

\[
\begin{pmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{pmatrix}
\]

\[
\sum_{v_j \in C_i} \deg V(v_j) \geq |E| / n
\]

|EI|/n=12/3=4.
Solving The Mapping Problem

\[
\begin{array}{cccccccc}
A & B & C \\
1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{array}
\]

\[
\sum_{v_j \in C_i} \deg V(v_j) \geq |E|/n
\]

IEI/n=12/3=4.
C1=red  \( \deg V(v1)=1 \leq 4 \).
\( \deg V(v1)+\deg V(v2)=3 \leq 4 \).
\( \deg V(v1)+\deg V(v2)+\deg V(v5)=5 \geq 4 \)

C1=1,2,5
Solving The Mapping Problem

\[ \begin{pmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{pmatrix} \]

\[ \sum_{v_j \in C_i} \deg V(v_j) \geq \frac{|E|}{n} \]

|EI/n=12/3=4.
C1=1,2,5
C2=green \( \deg V(v3) + \deg V(v4) = 4 = 4 \)
C2=3,4 |
Solving The Mapping Problem

\[
\begin{bmatrix}
A & B & C \\
1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

\[
\sum_{v_j \in C_i} \deg V(v_j) \geq |E| / n
\]

\(\text{IEI}/n=12/3=4.\)
\(C1=\{1,2,5\} \quad C2=\{3,4\}\)
\(C3=\text{blue} \quad \text{We have } v6, v7 \text{ for set blue.}\)
\(C3=\{6,7\}\)
Solving The Mapping Problem

1, 2, 5  3, 4  6, 7

A B C

1-A  2-A  2-B  5-A  5-C
1 2 3 4 5
6 7

3, 4

6, 7

1-A
2-A
2-B
5-A
5-C

3-B
4-A
4-B
4-C

6-B
6-C
7-C
Solving The Mapping Problem

fp1 has 3 stalls to read edges falling on A
1-A, 2-A, 4-A, 5-A
Solving The Mapping Problem

fp2 has 0 stalls to read edges falling on B
2-B, 3-B, 4-B, 6-B
Solving The Mapping Problem

fp3 has 0 stalls to read edges falling on C
6-C, 7-C, 5-C, 4-C
Solving The Mapping Problem
Solving The Mapping Problem

A space is a color \( i \) such that
\[ C_i \cap Nbr(f_j) = \emptyset \]

For \( Nbr(A) \)

\[
e_j = 1 + (r_{\max} - 1)n + d_{\min} - |Nbr(f_j)|
\]
Solving The Mapping Problem

\[ \text{Cost} \left( v_i \right) = \sum_{f_j \in Nbr \left( v_i \right)} e_j^2 \]

\( e_j \) is the number of spaces in the shortest path along the \( n \) - color circle that connects all \( v_i \in Nbr \left( f_j \right) \).
Solving The Mapping Problem

$$\text{gain}(v_i) = Cost(v_i) \mid -Cost(v_i)$$

\( v_i \in C_1 \mid v_i \in C_2 \)
Solving The Mapping Problem

\( C_1 \rightarrow C_3 \)

\( C_1 = 1, 2, 5 \quad C_2 = 3, 4 \quad C_3 = 6, 7 \quad Nbr(v_1) = A \)

\[
\text{Cost } (v_1) = \sum_{f_j \in Nbr(v_1)} e^2_j = e^2_A \quad e_A = 1 + (r_{\text{max}} - 1)n + d_{\text{min}} - |Nbr(f_A)|
\]

\( Nbr(A) = 1, 2, 4, 5 = r_{\text{max}} = 3 \ (1, 2, 5 \text{ are red}) \)

\( d_{\text{min}} = 0 \)

\( e_A = 1 + (3 - 1) \cdot 3 - 4 = 3 \)
Solving The Mapping Problem

\[ C_1 = 2, 5 \quad C_2 = 3, 4 \quad C_3 = 6, 7, 1 \]

\[ \text{Nbr}(A) = 1, 2, 4, 5 \]

\[ r_{\text{max}} = 2 \]

\[ d_{\text{min}} = 0 \]

\[ e_A = 1 + (2 - 1) \cdot 3 + -4 = 0 \]

\[ \text{gain}(v_1) = 3^2 - 0 = 9 \]
Solving The Mapping Problem

\[ Nbr(f_j) = \{v_1, v_2, v_3, v_4\} \]
Solving The Mapping Problem

\[ \text{Nbr}(A) \]
Solving The Mapping Problem

\[ i=1 \rightarrow e_j = 0 \]
Solving The Mapping Problem

- $i=1 \rightarrow e_j = 0$
- $i=2 \rightarrow e_j = 1$
Solving The Mapping Problem

- $i=1 \rightarrow e_j = 0$
- $i=2 \rightarrow e_j = 1$
- $i=3 \rightarrow e_j = 1$
Solving The Mapping Problem

- $i=1 \rightarrow e_j = 0$
- $i=2 \rightarrow e_j = 1$
- $i=3 \rightarrow e_j = 1$
- $i=4 \rightarrow e_j = 2$
Solving The Mapping Problem

- i=1 $\rightarrow$ $e_j = 0$
- i=2 $\rightarrow$ $e_j = 1$
- i=3 $\rightarrow$ $e_j = 1$
- i=4 $\rightarrow$ $e_j = 2$
- i=5 $\rightarrow$ $e_j = 3$
Solving The Mapping Problem

- i=1 → e_j = 0
- i=2 → e_j = 1
- i=3 → e_j = 1
- i=4 → e_j = 2
- i=5 → e_j = 3
- i=6 → e_j = 3
Solving The Mapping Problem

\[ f_A, f_B, f_C \in Nbr(v_i) \]
\[ \cos t(v_i) = e_A + e_B + e_C \]

\[ e_A = 1, e_B = 1, e_C = 1 \]
Solving The Mapping Problem

\[ f_A, f_B, f_C \in Nbr (v_i) \]

\[ \cos t(v_i) = e_A + e_B + e_C \]

\[ e_A = 3, e_B = 0, e_C = 0 \]
Solving The Mapping Problem
Solving The Mapping Problem

\[(r_{\text{max}} - 1)n\]
Solving The Mapping Problem

\[(r_{\text{max}} - 1).n\]
Solving The Mapping Problem

\[(r_{\text{max}} - 1).n + d_{\text{min}}\]
Solving The Mapping Problem

\[ 1 + (r_{\text{max}} - 1)n + d_{\text{min}} - |Nbr(f_j)| = e_j \]
Solving The Mapping Problem

How many times do we calculate $e_j$?

$$\sum_{j=0}^{K} \sum_{i=1}^{K-j} 2 \cdot \text{deg} V(v_i) \approx \frac{K(K-1)}{2} \times 2 \times \text{deg} V_{ave}$$

where $K = |C_1 \cup C_2|$
Solving The Mapping Problem

e_j \text{ depends on only on } |\text{Nbr } (f_j) \cap C_1| \text{ and } |\text{Nbr } (f_j) \cap C_2|

S_j = \text{Nbr } (f_j) \cap \{C1 \cup C2\} , \quad 1_j = |S_j|

s_j = |\text{Nbr } (f_j) \cap C_1| , \quad |\text{Nbr } (f_j) \cap C_2| = 1_j - s_j
Solving The Mapping Problem

\[
\text{gain} \left( v_i \right) = \sum_{f_j \in \text{Nbr} \left( v_i \right)} \left( e_j^2 \mid_{\text{vi} \in C_1} - e_j^2 \mid_{\text{vi} \in C_2} \right) = \sum_{f_j \in \text{Nbr} \left( v_i \right)} \text{gain} \left( v_i, f_j \right)
\]

\[
\text{gain} \left( v_i, f_j \right) = e_j^2 \left( s_j \right) - e_j^2 \left( s_j - 1 \right) \quad \text{for} \quad v_i \in \left\{ \text{Nbr} \left( f_j \right) \cap C_1 \right\}
\]

update \quad s_j

s_j = s_j - 1 \quad \text{if} \quad v_i \in \text{Nbr} \left( f_j \right) \text{ is moved from } C_1 \text{ to } C_2
Solving The Mapping Problem

\[ \text{gain}(v_i, f_j) = e_j^2(s_j) - e_j^2(s_j + 1) \quad \text{for} \quad v_i \in \{Nbr(f_j) \cap C_2\} \]

update \quad s_j

\[ s_j = s_j + 1 \quad \text{if} \quad v_i \in Nbr(f_j) \text{ is moved from } C_1 \text{ to } C_2 \]
Solving The Mapping Problem

\[ K \times \text{deg} V_{\text{ave}} \times 2 \times (\text{deg} F_{\text{max}} + 1) \]

\[ TMC_v = \sum_{f_j \in F} e_j^2 \quad , \quad TMC_f = \sum_{v_i \in V} e_i^2 \]
Solving The Scheduling Problem

- Linear ordering problem (NP hard)

1, 3, 5, 6, 7, 8, 9

variable \( \text{processor}_i \)
Solving The Scheduling Problem

- Linear ordering problem (NP hard)

1,3,5,6,7,8,9

\[
\begin{align*}
\deg(v_1) &= 5 & \deg(v_7) &= 3 \\
\deg(v_3) &= 2 & \deg(v_8) &= 3 \\
\deg(v_5) &= 3 & \deg(v_9) &= 3 \\
\deg(v_6) &= 3
\end{align*}
\]

variable \quad processor \: _i
Solving The Scheduling Problem

- Linear ordering problem (NP hard)

1, 3, 5, 6, 7, 8, 9

\[
\begin{array}{c}
 v_3 \\
v_5 \\
v_6 \\
v_7 \\
v_8 \\
v_9 \\
v_1
\end{array}
\]

\[\begin{align*}
\text{deg}(v_1) &= 5 & \text{deg}(v_7) &= 3 \\
\text{deg}(v_3) &= 2 & \text{deg}(v_8) &= 3 \\
\text{deg}(v_5) &= 3 & \text{deg}(v_9) &= 3 \\
\text{deg}(v_6) &= 3
\end{align*}\]
Solving The Scheduling Problem
Solving The Scheduling Problem
Solving The Scheduling Problem

get maximum decrease in cost
Solving The Scheduling Problem
Solving The Scheduling Problem

compute maximum weight matching
Solving The Scheduling Problem

compute maximum weight matching
Solving The Scheduling Problem

compute maximum weight matching

apply the process again

to get maximum decrease

until no more compaction can be made
Solving The Scheduling Problem

repeat whole process

until no more improvement can be made
The system

Each processor can write to a single memory bank only.

\[ P_j M_i = \delta_{i,j} = \begin{cases} 
1 & \text{if } i=j \\
0 & \text{otherwise} 
\end{cases} \quad \forall j \]
The system

Each memory bank can be written at most once each cycle $t$

$M_{w_{i,t}} = 1$ if memory bank $i$ is written at cycle $t$

$M_{w_{i,t}} = 0$ otherwise

$$\sum_{i=1}^{n} M_{w_{i,t}} \leq n \quad \forall t$$
The system

Each memory bank can be read at most once each cycle \( t \)

\[
Mr_{i,t} = 1 \quad \text{if memory bank } i \text{ is read at cycle } t
\]

\[
Mr_{i,t} = 0 \quad \text{otherwise}
\]

\[
\sum_{i=1}^{n} Mr_{i,t} \leq n \quad \forall t
\]
The system

- Each processor produces at most one data element each cycle.

\[ P_{i,t} = \begin{cases} 1 & \text{if processor } i \text{ produces data element at cycle } t \\ 0 & \text{otherwise} \end{cases} \]

\[ \sum_{i=0}^{n-1} P_{i,t} \leq n \quad \forall t \]
The system

- Each processor consumes at most one data element each cycle.

\[ C_{i,t} = 1 \text{ if processor } i \text{ consumes data element at cycle } t \]
\[ C_{i,t} = 0 \text{ otherwise} \]
\[ \sum_{i=0}^{n-1} C_{i,t} \leq n \]
The system

Each processor reads cyclically from memory

\[ P_{jM}(t) = \begin{cases} 
1 & \text{if processor } j \text{ reads from memory bank } i \text{ at cycle } t \\
0 & \text{otherwise} 
\end{cases} \]

\[ \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} P_{jM}(t) = n \quad \forall t \]

\[ P_{jM}(t \mod n - 1) \neq P_{jM}(t + 1 \mod n - 1) \]
The system

- If processor $p_j$ decides to start processing node $n_i$, it must read all edges falling on $n_i$, before it starts another node, at most one edge at each cycle continuously, but in any order.
Thank you for listening !!!