Energy Efficient Analog and Digital Iterative Error-Correcting Decoders

Saied Hemati, Ph.D., SMIEEE
Introduction
- Channel coding
- Low-density parity-check (LDPC) codes
- Iterative decoding algorithms

Analog iterative decoders
- Dynamics of analog iterative decoders
- An analog min-sum iterative decoder

Novel digital iterative decoders
- Low-power min-sum iterative decoder
- Low-complexity differential binary message passing decoder
- Stochastic decoder for IEEE 802.3an standard
- Stochastic Chase decoding, a fault tolerant architecture
- RHS
- ITU-T G. 975.1
- .....
- Channel coding is used to increase reliability against noise.
- In 1948, Shannon introduced channel capacity concept.
- Channel coding is accomplished by introducing redundancy and it is widely used in modern communication systems.
- Stochastic nature of nano-devices makes design for worst-case-scenario too expensive. Channel coding concept can be applied to this problem.
In 1993, turbo codes were discovered and became the coding scheme of choice in late 1990s.

In 1997, David MacKay found that low-density parity-check (LDPC) codes are also capacity achieving codes.

LDPC codes were originally developed by Gallager in 1960s, but were forgotten as they seemed to be impractical to implement.

LDPC codes have been chosen in a few recent standards:
- IEEE 802.16e (WiMAX 35Mbps Uplink/ 144Mbps Downlink updated to 1Gbps)
- IEEE 802.3an (10Gbps Ethernet over CAT6 cables)
- Broadband Satellite Digital Video Broadcasting (DVB-S2), 2nd generation
- ITU-T G.975.1 (40-100 Gbps optical communication)
- ITU-T G.9960 (home networking over power line, coaxial cable, and phone line, up to 1Gb/s)
message \( (m) \)
\[
\begin{bmatrix}
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0
\end{bmatrix}
\]

Codeword \( (c) \)

\[
c_i \oplus c_j \oplus c_l = 0
\]

\[
c_{1\times n} = m_{1\times k} G_{k\times n}
\]

\[
c_{1\times n} H^{T}_{(n-k)\times n} = 0
\]

Low-density parity-check codes have sparse \( H \) matrices.
Each received bit has some information about its neighboring bits. In iterative decoding, messages are passed iteratively through edges to obtain a more reliable estimation for each bit.

\[ c_i \oplus c_j \oplus c_l = 0 \implies \begin{cases} c_i = c_j \oplus c_l \\ c_j = c_i \oplus c_l \\ c_l = c_i \oplus c_j \end{cases} \]
Decoding performance improves when instead of one-bit inputs (hard inputs), probabilities corresponding to receiving a one or a zero is known (soft inputs).

Near optimum error correcting performance is observed when LDPC codes are decoded by soft-input iterative decoders.

Graph representation (Tanner graph) of an LDPC code depicts the structure of the corresponding iterative decoder.

Sum-product and min-sum decoding algorithms are the best known iterative decoding algorithms.
Digital iterative decoder chips are very large and power consuming.
- Floating-point operations (addition/multiplication, addition/log/tanh, addition/min)
- Number of edges
- Number of wires per edge
- Iterative nature of the decoder

Blanksby et al. reported a fully parallel digital iterative decoder for a (1024, 512) LDPC code.
- Power dissipation (SNR 0.3dB): 2W-0.69W
- Data throughput: 500Mbps
- Die size: 52.5 mm^2
- Technology: 0.16μm, 5LM
- Utilization factor: 50%
- Wire per edge (bits/message) = 4
- Number of wires: 26,624
- Gate count: 2.7M
Inspired by analog neural networks, a few groups designed low-power weakly inverted MOS/high-speed BiCMOS analog sum-product iterative decoder chips.

Analog sum-product decoders used Gilbert differential multiplier.

We found that implementing iterative decoders by analog circuits improves the dynamics of the iterative decoder.

We designed an MOS analog min-sum iterative decoder chip that works above and below threshold.
variable node: \( X = f(Y, R) \)

parity-check node: \( Y = g(X) \)

\[
\begin{cases} 
X = f(Y, R) \\
Y = g(X) 
\end{cases} \rightarrow X = f(g(X), R) = h(X, R)
\]

Successive substitution in iterative decoding: \( X_{l+1} = h(X_l, R) \quad l \geq 0 \)

\( X \): VAR outgoing messages

\( Y \): VAR incoming messages

\( R \): Received word

\( f \): Variable node operation

\( g \): parity-check operation
A simple model for analog continuous-time decoding:

\[ h(X, R) = f(g(X), R) \]

\[ \tau \dot{X}(t) = f(g(X(t)), R) - X(t) = h(X(t), R) - X(t) \]
Based on the forward Euler method and by assuming equal time constants:

\[ X(t + \Delta t) = X(t) + \beta \left( h(X(t), R) - X(t) \right), \quad \beta = \frac{\Delta t}{\tau} \]

In the discrete-time domain:

\[ X_{l+1} = X_l + \beta \left( h(X_l, R) - X_l(t) \right) \]

This is known as successive relaxation method.

Relaxation improves the performance of iterative decoders!
Throughput and convergence speed of analog decoders is very difficult to estimate. We showed by estimating the average time constant of the analog decoder, it is possible to predict the throughput and convergence of speed based on the decoders’ settling behavior.


Decoding performance of analog decoder is different for different message representation domains ($p$, $LR$, $LLR$, $\delta$). We analytically proved that dynamic equations is a function of message representation domain.


We showed that the analog min-sum decoder can be modeled as a piecewise linear system. We analytically solved the dynamic equations and found the eigenvalues. We also showed that for codes that have cycle in their structure, the corresponding dynamic equation becomes singular. We also linked singularity issue to trapping set that cause error floor in high signal to noise ratio region.


We improved the performance of sum-product algorithm that is the most powerful iterative decoder (Damped substitution).

Basic operations in min-sum decoder:

\[
m^{(l)}_{V \rightarrow C} = m_{V} + \sum_{C' \in N_{V} \setminus \{C\}} m^{(l-1)}_{C' \rightarrow V}
\]

\[
M^{(l)}_{V} = m_{V} + \sum_{C' \in N_{V}} m^{(l-1)}_{C' \rightarrow V}
\]

\[
m^{(l)}_{C \rightarrow V} = \left( \prod_{V' \in N_{C} \setminus \{V\}} \text{sign}(m^{(l)}_{V' \rightarrow C}) \right) \min_{V' \in N_{C} \setminus \{V\}} (|m^{(l)}_{V' \rightarrow C}|)
\]
Variable nodes

\[
\begin{align*}
    m_{V_1 \rightarrow C_1} &= m_{V_1} + m_{C_2 \rightarrow V_1} + m_{C_4 \rightarrow V_1} \\
    m_{V_1 \rightarrow C_2} &= m_{V_1} + m_{C_1 \rightarrow V_1} + m_{C_4 \rightarrow V_1} \\
    m_{V_1 \rightarrow C_4} &= m_{V_1} + m_{C_1 \rightarrow V_1} + m_{C_2 \rightarrow V_1} \\
    M_{V_1} &= m_{V_1} + m_{C_1 \rightarrow V_1} + m_{C_2 \rightarrow V_1} + m_{C_4 \rightarrow V_1}
\end{align*}
\]
Check nodes

In1
In2
In3
In4

Out1
Out2
Out3
Out4

|u| |u| |u| |u|

sign sign sign sign

min min min min

X X X X

1 2 3 4

5 6 7 8

16/35
It is often possible to convert a current mirror into a current-mode maximum winner-take-all (WTA) circuit.
A current-mode cascode 3-input min WTA.
Simulation results for the min WTA circuit
A low-voltage max WTA circuit
Simulation results for the low-voltage max WTA circuit
Architecture and microphotograph of the implemented chip
### CMOS digital (top 2) and analog (bottom 5) iterative decoders.

†: data throughput based on single measurement. ‡: data throughput based on the decoder’s average settling time.

---

<table>
<thead>
<tr>
<th>Code</th>
<th>CMOS Technology</th>
<th>Data Throughput Mb/s</th>
<th>Core (mm²)</th>
<th>Average Power (mW)</th>
<th>Energy Efficiency (nJ/b)</th>
<th>Number of Transistors or Gates</th>
<th>Loss in Coding Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1024, 512) LDPC Code</td>
<td>0.16µm 1.5V</td>
<td>500</td>
<td>52.5</td>
<td>690 @ 2.5dB 2000 @ 0dB</td>
<td>1.4 @ 2.5dB 4 @ 0dB</td>
<td>1,750K Gates</td>
<td>0.2dB</td>
</tr>
<tr>
<td>(2048, k) LDPC Code</td>
<td>0.18 µm 1.8V</td>
<td>320</td>
<td>14.3</td>
<td>787</td>
<td>2.46</td>
<td>220K Gates</td>
<td>0.05dB</td>
</tr>
<tr>
<td>Turbo Code (16-bit)</td>
<td>0.35µm 3.3V</td>
<td>13.3†</td>
<td>1.32</td>
<td>185 chip</td>
<td>13.9</td>
<td>-</td>
<td>1.1dB</td>
</tr>
<tr>
<td>(8,4) Hamming Tail-biting</td>
<td>0.5µm 3.3V</td>
<td>1†</td>
<td>0.083</td>
<td>1 core</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Turbo Code (40-bit)</td>
<td>0.35µm 3.3V</td>
<td>2†</td>
<td>4.1</td>
<td>6.8 core</td>
<td>3.4</td>
<td>30,000</td>
<td>0.5dB</td>
</tr>
<tr>
<td>(8,4) Hamming I-Trellis Graph II- Factor Graph</td>
<td>0.18µm 1.8V</td>
<td>I- 3.7† II- 3.7†</td>
<td>I- 0.002 II- 0.02</td>
<td>I- 0.15 II- 0.807 core</td>
<td>I- 0.042 II- 0.218</td>
<td>I- 500 II- 2,000</td>
<td>I- 1dB II- 0.6dB</td>
</tr>
<tr>
<td>(32,8) LDPC Code This Work</td>
<td>0.18µm 1.8V</td>
<td>6 ‡ (20)</td>
<td>0.57</td>
<td>5 chip</td>
<td>0.83 (0.25)</td>
<td>18,800</td>
<td>0.3dB (1dB)</td>
</tr>
</tbody>
</table>

---


Low-power digital min-sum decoder
- one wire per edge
- one 0→1 transition per iteration
- simple processing nodes
- pulse width approach is used for representing messages

(a) An example of representing magnitude of messages by a sequence of ‘1’s and finding minimum among these messages
(b) a degree four parity-check node.
A variable node of degree four

A 4-bit data circular shift register

A 4-bit sign circular shift register

Novel Digital Iterative Decoders
<table>
<thead>
<tr>
<th></th>
<th>This work</th>
<th>[5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>PWM-OMS</td>
<td>Bit-serial approx. MS</td>
</tr>
<tr>
<td>LDPC code</td>
<td>(660, 484)</td>
<td>(660, 484)</td>
</tr>
<tr>
<td>Process and supply voltage</td>
<td>0.13μm / 1.2V</td>
<td>0.13μm / 1.2V</td>
</tr>
<tr>
<td>Configuration</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Quantization width [bits]</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Pipelining</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Clock cycles / iteration</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Clock frequency [MHz]</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>Core area [mm²]</td>
<td>5.76</td>
<td>4.50</td>
</tr>
<tr>
<td>Cell area [mm²]</td>
<td>4.24</td>
<td>3.31</td>
</tr>
<tr>
<td>Gate equivalent [Kgates]</td>
<td>556</td>
<td>434</td>
</tr>
<tr>
<td>Max. iterations</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Coding gain [dB]α</td>
<td>0.5</td>
<td>-0.1</td>
</tr>
<tr>
<td>Avg. information throughput (5.5 dB) [Gbps]</td>
<td>5.71</td>
<td>9.18</td>
</tr>
<tr>
<td>Avg. power (5.5 dB) [mW]</td>
<td>376</td>
<td>465</td>
</tr>
<tr>
<td>Avg. energy per decoded bit (5.5 dB) [pJ/bit]</td>
<td>65.9</td>
<td>50.7</td>
</tr>
<tr>
<td>T/P per unit area [Gbps / mm²]</td>
<td>0.99</td>
<td>2.04</td>
</tr>
<tr>
<td>T/P per unit area per unit power [Gbps / (mm² - W)]</td>
<td>2.64</td>
<td>4.39</td>
</tr>
</tbody>
</table>

α Relative to conventional min-sum with q = 4 and 15 max. iterations.

Implementation results for a (660, 484) regular LDPC code

Novel Digital Iterative Decoders

Measurement Results
Low-complexity differential decoding binary message passing (DD-BMP)
- inspired by the dynamics of analog iterative decoders
- one wire per edge
- parity-check nodes are simple XOR gates
- variable nodes consist of several up/down counters that are initialized by the channel input and count received bits (1,-1). Outgoing messages are signs of these counters.

A degree four parity-check node

A degree four variable node


<table>
<thead>
<tr>
<th>Decoding algorithm</th>
<th>DD-BMP (273, 191)</th>
<th>MDD-BMP (273, 191)</th>
<th>MDD-BMP (1023, 781)</th>
<th>MDD-BMP (4095, 3367)</th>
<th>Split-row MS (2048, 1723)</th>
<th>Offset MS (2048, 1723)</th>
<th>Hybrid SBF (1057, 813)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDPC code</td>
<td>(17, 17)</td>
<td>(17, 17)</td>
<td>(32, 32)</td>
<td>(64, 64)</td>
<td>(6, 32)</td>
<td>(6, 32)</td>
<td>(33, 33)</td>
</tr>
<tr>
<td>Node degrees (d_v, d_c)</td>
<td>4641</td>
<td>4641</td>
<td>32736</td>
<td>262080</td>
<td>12288</td>
<td>12288</td>
<td>34881</td>
</tr>
<tr>
<td>Edge count</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td>65nm</td>
<td>65nm</td>
<td>65nm</td>
<td>65nm</td>
<td>65nm</td>
<td>65nm</td>
<td>180nm</td>
</tr>
<tr>
<td>Quantization bits</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Area</td>
<td>1.44mm²</td>
<td>0.276mm²</td>
<td>1.38mm²</td>
<td>15.37mm²</td>
<td>4.84mm²</td>
<td>5.35mm²</td>
<td>7.4mm²</td>
</tr>
<tr>
<td>Utilization</td>
<td>89%</td>
<td>90%</td>
<td>93%</td>
<td>43%</td>
<td>97%</td>
<td>84.5%</td>
<td>50%</td>
</tr>
<tr>
<td>Decoding iterations</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>11</td>
<td>8 + post proc.</td>
<td>40</td>
</tr>
<tr>
<td>Supply voltage (V)</td>
<td>1.0</td>
<td>0.6</td>
<td>1.0</td>
<td>0.6</td>
<td>1.3</td>
<td>1.2</td>
<td>0.7</td>
</tr>
<tr>
<td>Clock frequency (MHz)</td>
<td>400</td>
<td>122</td>
<td>550</td>
<td>168</td>
<td>180</td>
<td>55</td>
<td>700</td>
</tr>
<tr>
<td>Min. throughput (Gbps)</td>
<td>3.41</td>
<td>1.04</td>
<td>4.69</td>
<td>1.43</td>
<td>11.5</td>
<td>3.52</td>
<td>23.0</td>
</tr>
<tr>
<td>Av. throughput (Gbps)*</td>
<td>26.9</td>
<td>8.20</td>
<td>37.4</td>
<td>11.4</td>
<td>74.8</td>
<td>22.9</td>
<td>140.9</td>
</tr>
<tr>
<td>Av. power (mW)*</td>
<td>894</td>
<td>98.3</td>
<td>183</td>
<td>20.1</td>
<td>989</td>
<td>109</td>
<td>5354</td>
</tr>
<tr>
<td>Av. energy (pJ/bit)*</td>
<td>33.3</td>
<td>12.0</td>
<td>4.88</td>
<td>1.76</td>
<td>13.2</td>
<td>4.75</td>
<td>37.9</td>
</tr>
<tr>
<td>Scaled energy (pJ/bit)†</td>
<td>33.3</td>
<td>4.88</td>
<td>13.2</td>
<td>37.9</td>
<td>8.64</td>
<td>34.3</td>
<td>153.9</td>
</tr>
<tr>
<td>Scaled throughput (Gbps/mm²)†</td>
<td>18.7</td>
<td>135.5</td>
<td>54.2</td>
<td>9.2</td>
<td>19.2</td>
<td>10.6</td>
<td>1.1</td>
</tr>
</tbody>
</table>

* Averages are measured at \(E_b/N_0 = 4.5\) dB in this work, at \(E_b/N_0 = 4.55\) dB in [7], at \(E_b/N_0 = 5.5\) dB in [9], and at BER = 10\(^{-5}\) in [15].
† Average throughput per unit area scaled to 65nm.
Stochastic Decoder for IEEE 802.3an (10GBASE-T)
- A (2048,1723) LDPC code.
- Variable nodes of degree 6, and check nodes of degree 32, with over 12,000 edges.
- We designed a majority-based TFM stochastic decoder.
- Core area 6.38 mm² in CMOS 90 nm technology
- Maximum throughput of 57.1 Gb/s
<table>
<thead>
<tr>
<th></th>
<th>This work</th>
<th>[36],[38] (U of T)</th>
<th>[35,37] (UC Berkeley)</th>
<th>[33] (UC Davis)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decoding Algorithm</td>
<td>MTFM-based Stochastic</td>
<td>Bit-serial Approximate MSA</td>
<td>Offset MSA+ post-</td>
<td>MSA Split-16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>processing</td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td>Fully Parallel</td>
<td>Fully Parallel</td>
<td>Partially Parallel</td>
<td>Fully Parallel</td>
</tr>
<tr>
<td>CMOS Technology</td>
<td>90nm</td>
<td>90nm</td>
<td>65nm</td>
<td>65nm</td>
</tr>
<tr>
<td>Input Quantization</td>
<td>6 bits</td>
<td>4 bits</td>
<td>4 bits</td>
<td>Not reported</td>
</tr>
<tr>
<td>Clock Frequency</td>
<td>500MHz</td>
<td>250MHz</td>
<td>700MHz</td>
<td>100MHz</td>
</tr>
<tr>
<td>Iteration</td>
<td>Max. 400</td>
<td>8</td>
<td>8+4 post-Processing</td>
<td>15</td>
</tr>
<tr>
<td>Clock per iteration</td>
<td>1</td>
<td>4</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>Decoding Latency</td>
<td>800ns</td>
<td>128ns</td>
<td>206ns</td>
<td>150ns</td>
</tr>
<tr>
<td>Core Area</td>
<td>6.38mm²</td>
<td>9.8mm²</td>
<td>5.35mm²</td>
<td>3.8mm²</td>
</tr>
<tr>
<td>Scaled Area per Decoded bit (in 90nm)</td>
<td>3115μm²</td>
<td>4785μm²</td>
<td>5008μm²</td>
<td>3557μm²</td>
</tr>
<tr>
<td>Throughput</td>
<td>57.1Gb/s</td>
<td>16Gb/s</td>
<td>47.7Gb/s</td>
<td>13.8Gb/s</td>
</tr>
</tbody>
</table>
Stochastic Chase Decoding, a Fault Tolerant Architecture

- Instead of a big soft decoder, we use many simple hard-decision decoders (HDDs).
- One example is soft-decoding of Reed-Solomon codes.
- The probabilistic search outperforms a deterministic search.
- This architecture is fault tolerant and remain intact if there is any soft error in HDDs.
- We may exchange reliability with power efficiency.
- This design is well suited for low-reliability nano-scale devices.

High-speed relaxed half-stochastic (RHS) decoder
- inspired by the dynamics of analog iterative decoders
- one wire per edge
- parity-check nodes are simple XOR gates
- messages are stochastic binary streams.
- can be pipelined easily
- latency can be divided by the number of decoders
- comparable with floating-point sum-product

(a) A degree four RHS check node (b) a degree four RHS variable node
Novel Digital Iterative Decoders

An LDPC Decoder for ITU-T G.975.1
- Throughput 40-100Gbps
- This is a (32643,30592) LDPC code.
- Variable nodes of degree 7, and check nodes of degree 111-112, with over 230,000 edges.
- Fully parallel implementation of binary message passing algorithm requires at least 250,000 wires.
- We designed a partially parallel scheme for implementing this decoder based on the structure of the code.