Combining Flexibility with Low Power: Dataflow and Wide-pipeline LDPC Decoding Engines in the Gbit/s Era

João Andrade*, Frederico Pratas†, Gabriel Falcão*, Vitor Silva*, Leonel Sousa†

* Instituto de Telecomunicações, DEEC, Universidade de Coimbra
† INESC-ID, Instituto Superior Técnico, University of Lisbon
Outline

LDPC Codes Fundamentals

Decoding Architectures
- Dataflow Decoder
- Wide-pipeline Decoder

Experimental Results
- Utilization and Throughput
- Energy Efficiency
- Bandwidth Analysis

Conclusion
Motivation

• FEC in communication systems require:
  • High-Throughput, Low Latency, Low Power
  • But, Turbo or LDPC decoders are complex solutions

• Dedicated accelerators are typically deployed
  • We could live without high NRE
  • We could also live without HDL

• Programmable logic targeted through high-level languages
  • FPGAs
  • MaxelerOS (JAVA)
  • OpenCL (C)
Soft-Decoding Algorithms
Message-passing I

Parity-Check Equations

\[ c_0 + c_3 + c_6 = 0 \]
\[ c_0 + c_1 + c_2 = 0 \]
\[ c_3 + c_4 + c_5 = 0 \]

Parity-Check Matrix

\[
H = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}
\]

Tanner Graph
• \(\gamma_n\) msgs from channel demod
• \(\alpha_{mn}\) msgs are init’d w/ \(\gamma_n\)
• BNs send \(\alpha_{mn}^{(0)}\) to CNs
• from \(\alpha_{mn}^{(0)}\) CNs compute \(\beta_{mn}^{(0)}\) (CN proc.)
• CNs send \(\beta_{mn}^{(0)}\) to BNs
• BNs compute \(\alpha_{mn}^{(1)}\) from \(\beta_{mn}^{(0)}\) (BN proc.)
• BNs also compute \(\hat{\alpha}_n^{(1)}\) and their bit state
• Repeat until decoded or max. iter. reached
Dataflow Decoder
Maxeler Dataflow Engine

- $m$ cores are defined per decoder
- $p$ decoders are defined per accelerator
- message-passing through the Tanner graph is optimized using BRAMs
Dataflow Decoder
Maxeler Dataflow Engine

- dispatching overhead is negligible for steady-state operation

- BN and CN mode operation in the core are seamlessly interchangeable and are not mutually exclusive
  - BN/CN execution order is rearranged wo/ penalty
  - leading BNs/CNs overlap computation with trailing CNs/BNs
  - \(~ 5\% \) of clock cycles saved each iteration
- data is streamed from each processing unit w/ reduced memory overhead
• Wide-pipeline decoder is constrained – OpenCL 1.0 compliant
• Memory hierarchy is OpenCL implementation dependent
Wide-pipeline Decoder

Altera OpenCL Decoder

- fewer logic resources available for the OpenCL kernels
- $p=1$ for all cases and $m=\{1, 2\}$-way SIMD
- work-item initiation interval of 1 clock cycle was achieved but clock frequency is low
- iterations are unrolled by assigning more or less work-items (not true-OpenCL model)

- $\text{__global memory} \rightarrow \text{SDRAM}$, $\text{__local memory} \rightarrow \text{BRAM}$
## Experimental Results

### FPGA Utilization

<table>
<thead>
<tr>
<th></th>
<th>Altera OpenCL¹ (Stratix V D5)</th>
<th>MaxCompiler² (MAX2 Virtex5 LX330T)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LUTs/LEs / FFs (%)</strong></td>
<td>(1,1) 28/23 41/36</td>
<td>(1,180) 12/16 19/25 33/44 62/82 19/25 33/43 33/44</td>
</tr>
<tr>
<td>BRAMs (%)</td>
<td>42</td>
<td>27 29 35 49 41 71 47</td>
</tr>
<tr>
<td>Clock (MHz)</td>
<td>240 157</td>
<td>300 260 200 175 250 180 150</td>
</tr>
<tr>
<td>Thr. (Mbit/s)</td>
<td>16 21</td>
<td>270 468 720 1260 450 648 540</td>
</tr>
<tr>
<td>Rel. Thr. (Mbit/s/core)</td>
<td>16 10.5 6</td>
<td>0.76 1.32 2.03 3.54 1.27 1.82 1.52</td>
</tr>
<tr>
<td>Min. Dec. Iter.</td>
<td>0.02 0.03</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MaxCompiler² (MAX3 Virtex6 SX475T)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LUTs/LEs / FFs (%)</strong></td>
<td>(1,45) 6/4 9/7 17/14 32/27 9/7 16/14 32/26 16/14 31/27 63/52 32/27 64/53</td>
</tr>
<tr>
<td>BRAMs (%)</td>
<td>6 6 8 13 10 19 37 12 22 43 15 24</td>
</tr>
<tr>
<td>Clock (MHz)</td>
<td>260 260 200 175 250 200 150 150 90 170 80</td>
</tr>
<tr>
<td>Thr. (Mbit/s)</td>
<td>234 468 720 1260 450 720 1080 540 1080 1296 1224 1152</td>
</tr>
<tr>
<td>Rel. Thr. (Mbit/s/core)</td>
<td>5.2 5.2 4 3.5 5 4 3 3 3 1.8 3.4 3.6 3.4 3.4</td>
</tr>
<tr>
<td>Min. Dec. Iter.</td>
<td>0.66 1.32 2.03 3.54 1.27 2.03 3.04 1.52 3.04 3.65 3.44 3.24</td>
</tr>
</tbody>
</table>

¹ $N = 1944$ rate 1/2 WiFi code. ² $N = 64800$ rate 1/2 DVB–S2 code. * In the wide-pipeline case $p = K_{SIMD}, m = K_{CUs}$.

- Dataflow approach yields high throughputs from end-terminal use to base-station
- Wide-pipeline high logic utilization is overestimated in early resource estimation
- Core decoding efficiency is higher in the wide-pipeline approach
### Experimental Results

#### Energy Efficiency

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(p=2,m=45)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>765</td>
<td>(p=2,m=45)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>270</td>
<td>(p=4,m=45)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>270</td>
<td>(p=2,m=90)</td>
</tr>
<tr>
<td>Thr. for 10 iterations (Mbit/s)</td>
<td>304</td>
<td>340</td>
<td>765</td>
<td>270</td>
</tr>
<tr>
<td>Power Cons. (W)</td>
<td>87 (est.)</td>
<td>79</td>
<td>&lt;1 (est.)</td>
<td>450</td>
</tr>
<tr>
<td>Freq. (MHz)</td>
<td>876</td>
<td>700</td>
<td>5</td>
<td>648</td>
</tr>
<tr>
<td>Eff. (Mbit/J/iter)</td>
<td>35</td>
<td>43</td>
<td>9</td>
<td>540</td>
</tr>
<tr>
<td>Dev. Effort</td>
<td>hour ~ days arch.</td>
<td></td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Re-usable modules</td>
<td></td>
<td></td>
<td>months</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>none</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>days core</td>
<td></td>
</tr>
</tbody>
</table>

All power figures obtained with Hall sensor, except on the K20c which provides power usage registers (marked with *).

[^1]: N = 1944 rate 1/2 WiFi code.
[^2]: N = 64800 rate 1/2 DVB-S2 code.
[^3]: N = 1152 rate 1/2 WiMAX code.
Experimental Results

Bandwidth Analysis

Decoding Iterations/Codeword
Decoding Iterations/s
PCIe 2nd Gen x8
- Max 3 (1,45) 234 Mbit/s
- Max 2 (1,45) 270 Mbit/s
- Max 2, Max 3 (1,360) 1260 Mbit/s
- Max 2, Max 3 (2,180) 1224 Mbit/s
- Max 2, Max 3 (2,360) 1152 Mbit/s
- Max 2, Max 3 (8,45), Max 3 (4,90) 1080 Mbit/s
- Max 2 (4,45) 648 Mbit/s
- Max 2, Max 3 (2,90) 540 Mbit/s
- Max 2, Max 3 (1,90) 468 Mbit/s
- Max 2 (1,45) 270 Mbit/s

Decoding Iterations/Codeword
Decoding Iterations/s
PCIe 3rd Gen x8
- DDR3−SDRAM
- Altera (1,2) 21 Mbit/s
- Altera (1,1) 16 Mbit/s

SNR
- High
- Lower
Summary

- Dataflow approach delivers the best performance
  - Flexibility is not compromised by the HLS
  - PCIe bandwidth can pose computation bottlenecks

- Wide-pipeline approach is limited also by OpenCL
  - OpenCL 1.0 is not tuned to FPGAS and an ecosystem of FPGA extensions is still missing
  - OpenCL 2.0 will get interesting with streaming and pipes

Outlook

- Can OpenCL deliver in FPGAs what it has in GPU computing?
- Abstraction on top of physically distinct memory spaces?
Thank you for staying around until the last presentation

(Offline) Questions?