

FPGA-Based Design and Implementation of a Multi-Gbps LDPC Decoder

Alexios Balatsoukas-Stimming and Apostolos Dollas

Technical University of Crete
Dept. of Electronic and Computer Engineering

August 30, 2012
FPL 2012, Oslo

1 Introduction

2 Preliminaries

- Channel Model
- LDPC Codes
- Decoding Algorithm

3 Decoder Architecture

- Message Quantization
- Variable and Check Node Processing Units
- Pipelining
- Early Termination

4 Results & Conclusion

LDPC Codes

- LDPC codes: Forward Error Correction (FEC) codes that exhibit **excellent error correction performance**.
- Adopted by many present and future standards.
- Hardware-friendly due to **inherent parallelism** of decoding algorithms.
- FPGAs can support multiple standards and rates via **runtime reconfiguration**.

Channel Model

- Memoryless AWGN channel:

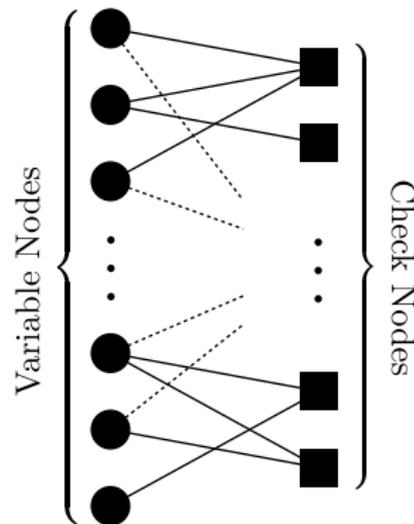
$$y_i = x_i + n_i, \quad n_i \sim \mathcal{N}(0, \sigma^2).$$

- Decoding consists of finding most likely x_i , based on \mathbf{y} , for all $i = 1, \dots, n$:

$$\hat{x}_i = \arg \max_{x_i} p(x_i | \mathbf{y}) \quad \leftarrow \text{NP-hard!}$$

LDPC Codes

- Defined through a **parity-check matrix H** .
- Represented by a **Tanner graph** →
- Decoding via **Min-Sum message passing** on the graph.



Min-Sum Decoding Algorithm

- Variable-to-check:
$$L_{ij} = \overbrace{2y_i/\sigma^2}^{\text{Initial LLR}} + \sum_{k \in \mathcal{C}(i)/j} R_{ki}.$$
- Check-to-variable:
$$R_{ij} = \left(\prod_{k \in \mathcal{V}(i)/j} \text{sign}(L_{ki}) \right) \min_{k \in \mathcal{V}(i)/j} |L_{ki}|.$$
- A maximum of k iterations is performed.

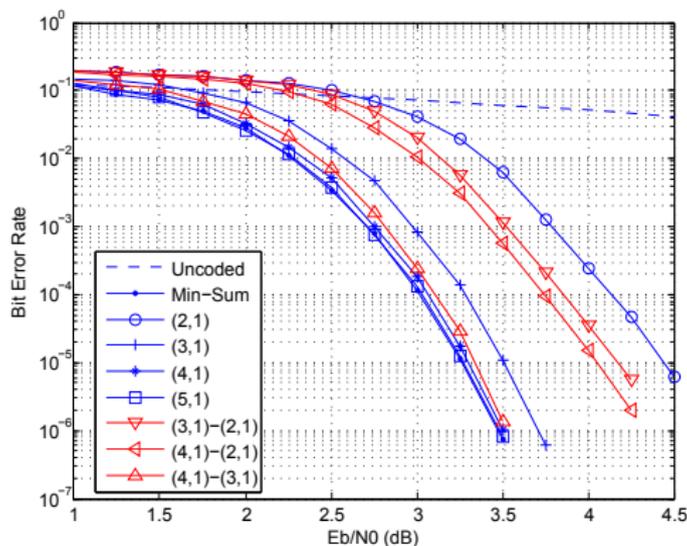
LDPC Decoder Architectures

- **Fully Parallel:** Every VN and CN represented in hardware.
 - **Very high** throughput.
 - **High** hardware utilization, **very complex** routing.
- **Serial:** One VN and one CN.
 - **Low** throughput.
 - **Very efficient** hardware utilization, **minimal routing**.
- **Partially Parallel:** **Compromise** between the two.

Message Quantization

- (n, m) signed fixed point quantization:
 - Total of n bits for each message.
 - m bits are used for fractional part.
- (n_1, m_1) – (n_2, m_2) **hybrid quantization**:
 - (n_1, m_1) quantization for initial LLR messages.
 - (n_2, m_2) quantization for variable-to-check and check-to-variable messages.
- By choosing $n_2 < n_1$, routing and processing unit complexity can be **significantly reduced**.

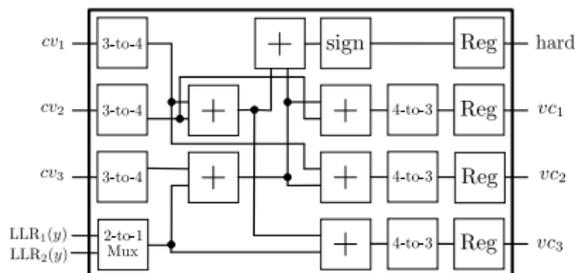
Effect of Message Quantization on Performance



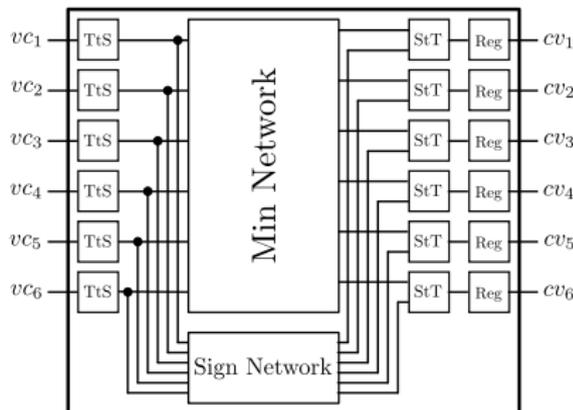
- Comparison of (4,1) with hybrid:
 - (4,1)-(3,1): **negligible** loss, **-25%** wires, **-45%** LUTs.
 - (3,1)-(2,1): **0.75 dB** loss, **-50%** wires, **-71%** LUTs.

Variable and Check Node Processing Units

- Straightforward **combinational logic** implementing the variable and check node update rules.



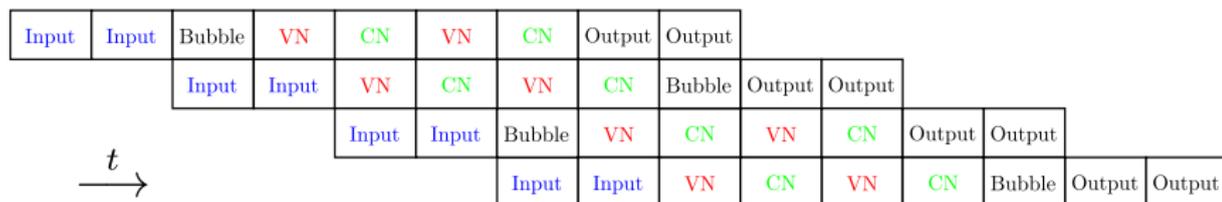
Variable node.



Check node.

Pipelining

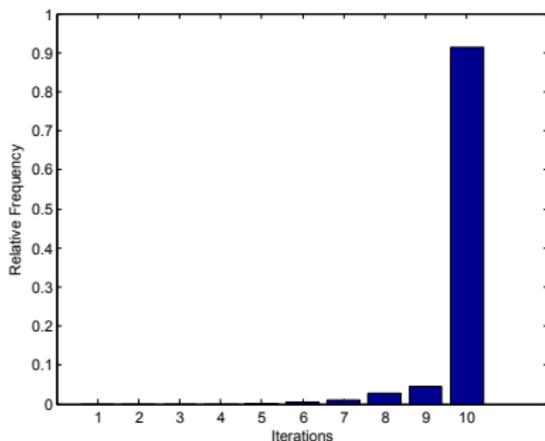
- Usually, one decoding iteration is considered to last one clock cycle → **high path delays**.
- **Idea**: add registers to reduce path delays.
- **Problem**: decoding now takes twice as many cycles.
- **Observation**: at each cycle, either VNs or CNs are idle.
- **Solution**: decode **two codewords simultaneously**.



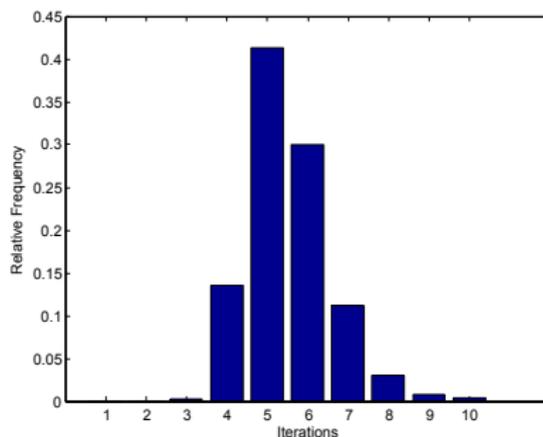
Early Termination

- If after some iteration we have reached a valid codeword, decoding can halt.
- At high E_b/N_0 , this can lead to **significant increase in average throughput**.
- **Significant I/O problems** due to non-uniform distribution of required iterations.

Early Termination



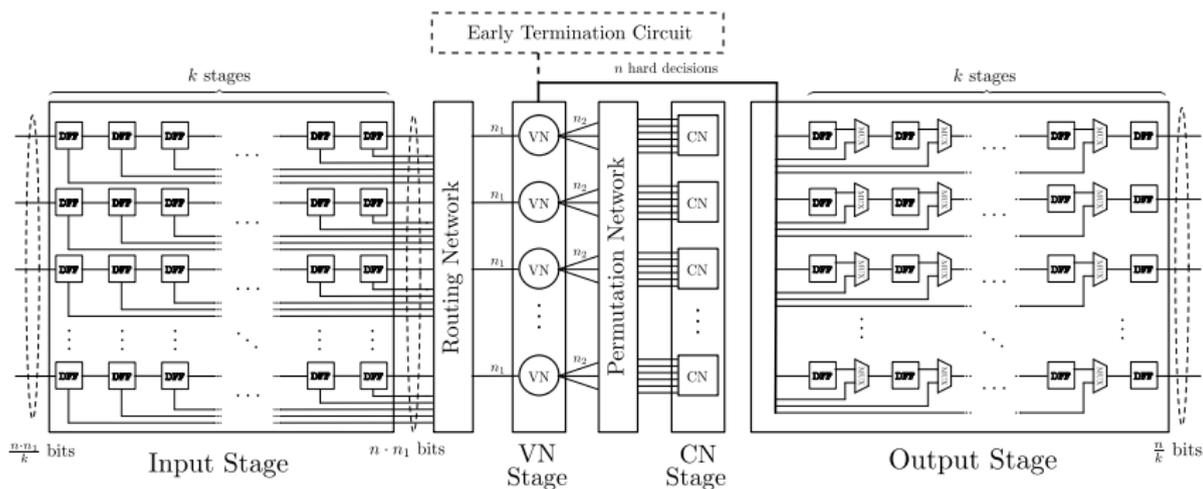
$E_b/N_0 = 2$ dB.



$E_b/N_0 = 3.5$ dB.

- **Idea:** force decoder to perform at least $k/2$ iterations.
 - **Small impact** on throughput.
 - **k guaranteed cycles** for I/O.

Overall Datapath



Results

	This work (4,1)–(3,1)	Chandraseetty & Aziz (2011)	gain/ loss
Decoding Algorithm	Min-Sum	MMS	
Clock Frequency	154.30 MHz	149.00 MHz	3.5%
Eb/N0 at 10^{-6} BER	3.5 dB	4 dB	0.50 dB
Av. Iter. at 10^{-6} BER	5.8	6.8	14.7%
Av. Throughput	14.6 Gbps	12.6 Gbps	15.9%
LUT Utilization	89.4%	98.5%	9.1%
Max. Delay	220 ns	121 ns	81.8%

Results

	This work (3,1)–(2,1)	Chandraseetty & Aziz (2011)	gain/ loss
Decoding Algorithm	Min-Sum	MMS	
Clock Frequency	211.40 MHz	149.00 MHz	41.9%
Eb/N0 at 10^{-6} BER	4.25 dB	4 dB	0.25 dB
Av. Iter. at 10^{-6} BER	5.6	6.8	17.6%
Av. Throughput	21.6 Gbps	12.6 Gbps	71.4%
LUT Utilization	47.6%	98.5%	50.9%
Max. Delay	161 ns	121 ns	33.1%

Conclusion

We presented an FPGA-based LDPC decoder architecture which:

- 1 Outperforms the state of the art by:
 - **15.9%** at a **0.50 dB lower** E_b/N_0 .
 - **71.4%** at a **0.25 dB higher** E_b/N_0 .
- 2 Requires **9.1%** and **50.9% less logic**, respectively.
- 3 **Fully addresses** the I/O problems due to early termination.

Thank you!

Questions?