λ-Min Decoding Algorithm of Regular and Irregular LDPC Codes

Frédéric Guilloud, Emmanuel Boutillon and Jean-Luc Danger

{frederic.guilloud, jean-luc.danger}@enst.fr, emmanuel.boutillon@univ-ubs.fr

Problems of LDPC decoder architecture :

- **Memory cost of Check-to-Variable (Lmn) and Variable-to-check messages (Zmn) .**
- **Complexity of the Check Node Processor for irregular codes.**

Solution: λ-Min Algorithm and its associated architecture

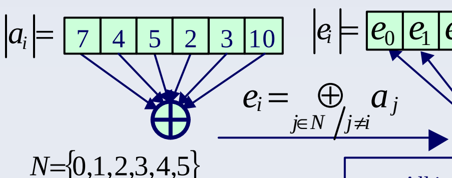
- ❑ **Sub-optimal algorithm to compute Zmn values.**
- ❑ **Local storage of a compressed version of Lmn messages.**
- ❑ **Serial computation of check node, with on the fly computation of Zmn messages.**

λ-Min Algorithm:

✓ **Idea:** Use the λ incoming messages which have the smallest magnitude, to compute extrinsic information.

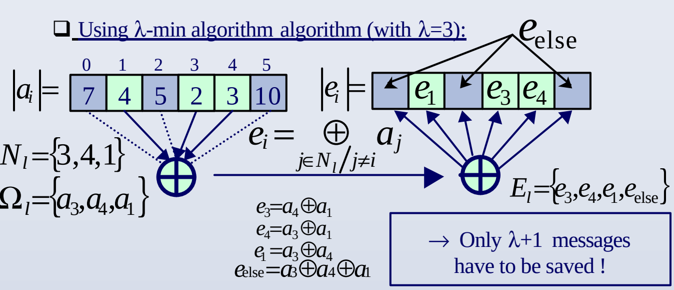
✓ **Example:** a check node of degree 6 with λ=3 (signs are omitted)

❑ Using BP algorithm:



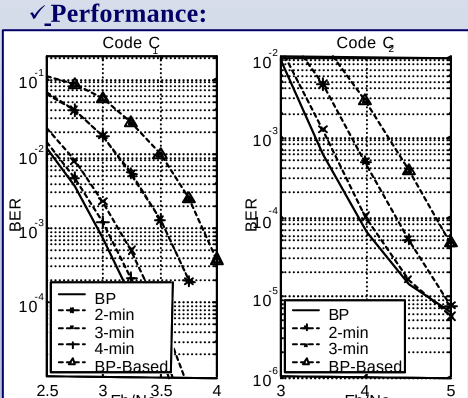
→ All information is required

❑ Using λ-min algorithm algorithm (with λ=3):



→ Only λ+1 messages have to be saved !

✓ **Performance:**



C1 : (5,10), N= 816 from http://www.inference.phy.cam.ac.uk/mackay/codes/data.html

C2 : Irregular rate 0.85 LDPC code, N=2000 (41~42 bits / PC) from http://lthcwww.epfl.ch/research/ldpcopt.

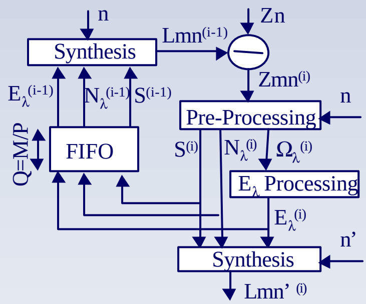
50 decoding iterations

Note: the addition of a correcting factor (offset) in operation ⊕ allows the 3-Min algorithm to over perform the BP algorithm

Architecture

Based on "decoder first code design" architecture (Boutillon et al., ISTC&RT'00)

- ❑P memory banks storing variables $Zn^{(i-1)}$ and processing $Zn^{(i)}$ by accumulation
- ❑P parity check processors working in parallel.
- ❑Shuffle (and unshuffle) network between memory banks and PC processors.



❑ Parity Check Processor:

The M parity checks are processed by P parity check processors in Q macro-cycle.

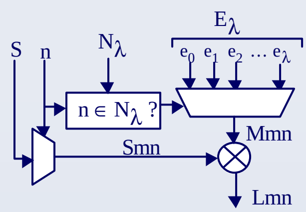
Each processor features 2 synthesis blocks and 1 pre-processing block

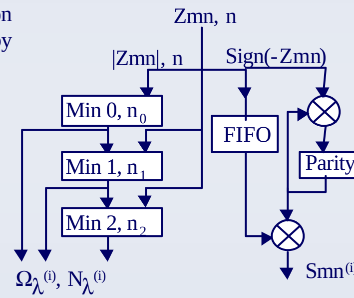
✓ **Pre-Processing:**

- Sign processing
- Sorting the λ minimum magnitude

✓ **Synthesis:**

Generate each extrinsic information from the λ+1 messages computed by the λ-min algorithm.





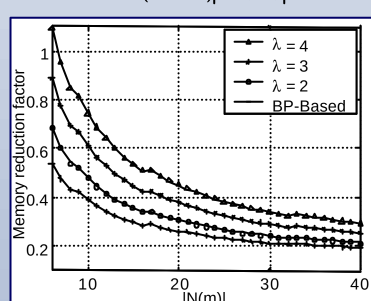
Memory Reduction

The ratio between the memory used by the λ-min algorithm and the memory used by the BP algorithm for one parity check is given by:

Results of the λ-min algorithm $E\lambda$ — λ indexes of $N_\lambda(m)$ — Sign of Lmn messages (Smn)

$$\frac{(\lambda+1)N_b + \lambda \log_2 |N(m)| + |N(m)|}{(N_b+1)|N(m)|}$$

BP algorithm results: |N(m)| different extrinsic values coded on Nb+1 bits



Arithmetic is made using Nb bits for magnitude and 1 bit for the sign ; the graph is made with Nb = 5.

For |N(m)|=20 and λ=3, up to 60 % of memory save compared to classical LDPC decoder.

3rd Inter. Symp. on Turbo Codes & Related Topics, Brest, France, 1 - 5 Sept. 2003