

An Efficient, Portable and Generic Library for Successive Cancellation Decoding of Polar Codes

Adrien Cassagne, Bertrand Le Gal, Camille Leroux, Olivier Aumage and Denis Barthou

Exploring Soft ECC Decoding

Growing interest for software Error Correction Codes implementations

- Leverage powerful, energy efficient procs.
- Reduce dev. cost and time to market
- Validate and optimize new algorithms

Recent **Successive Cancellation** soft decoders for Polar ECC codes strongly benefit from modern CPUs capabilities and SIMD units, open the way to a wide optimization range.

Introducing **P-EDGE**, an environment for exploring Polar ECC decoders.

- Specialized skeleton generator
- Building blocks library

Decoding of Polar Codes

The **Successive Cancellation (SC)** decoding algorithm: a depth-first binary tree traversal algorithm based on 3 key functions:

$$\begin{cases} f(\lambda_a, \lambda_b) &= \text{sign}(\lambda_a \cdot \lambda_b) \cdot \min(|\lambda_a|, |\lambda_b|) \\ g(\lambda_a, \lambda_b, s) &= (1 - 2s)\lambda_a + \lambda_b \\ h(s_a, s_b) &= (s_a \oplus s_b, s_b). \end{cases}$$

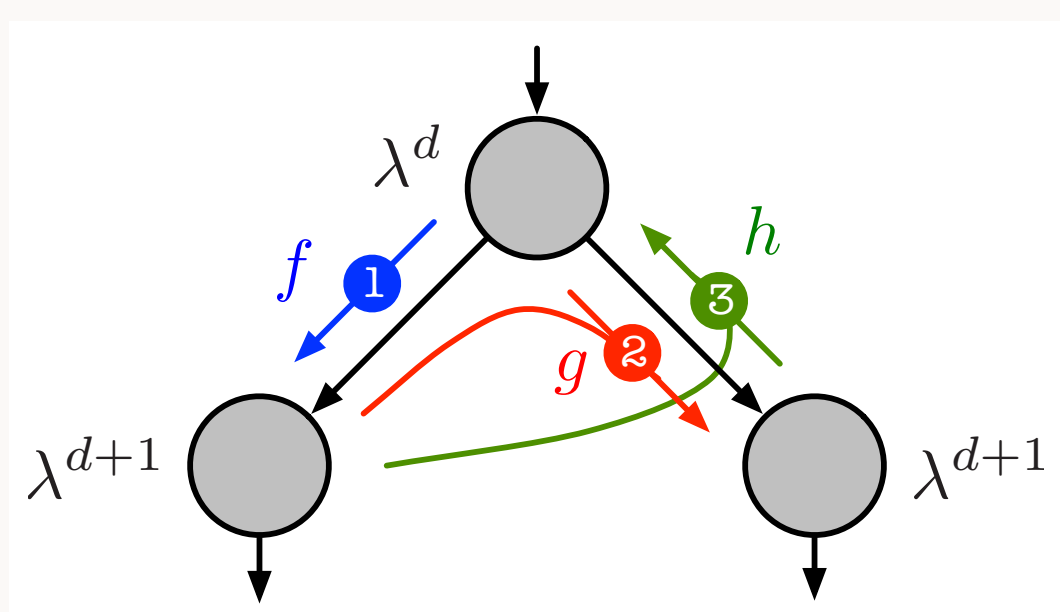
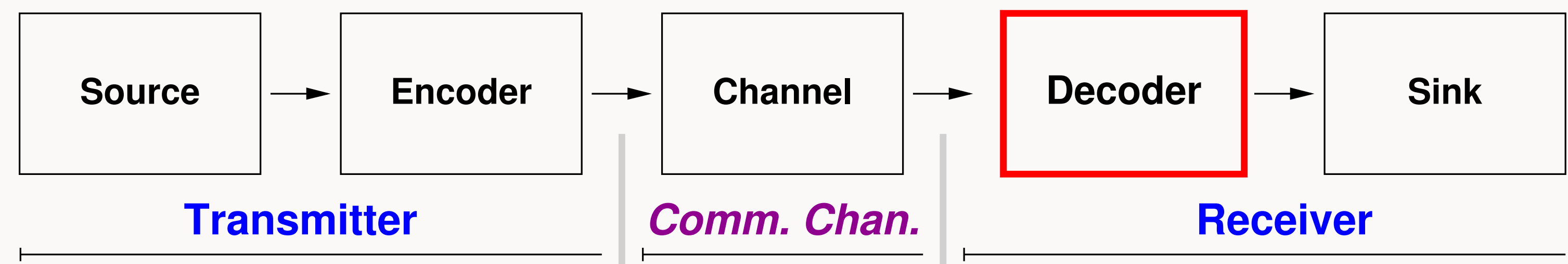


Figure 1: Per-node downward and upward computations

The Communication Chain



P-EDGE Generation Process

The set of the rewriting rules used by P-EDGE is shown Fig. 4: it enables to generate the source code of the decoders. Electronics scientist can enhance this set.

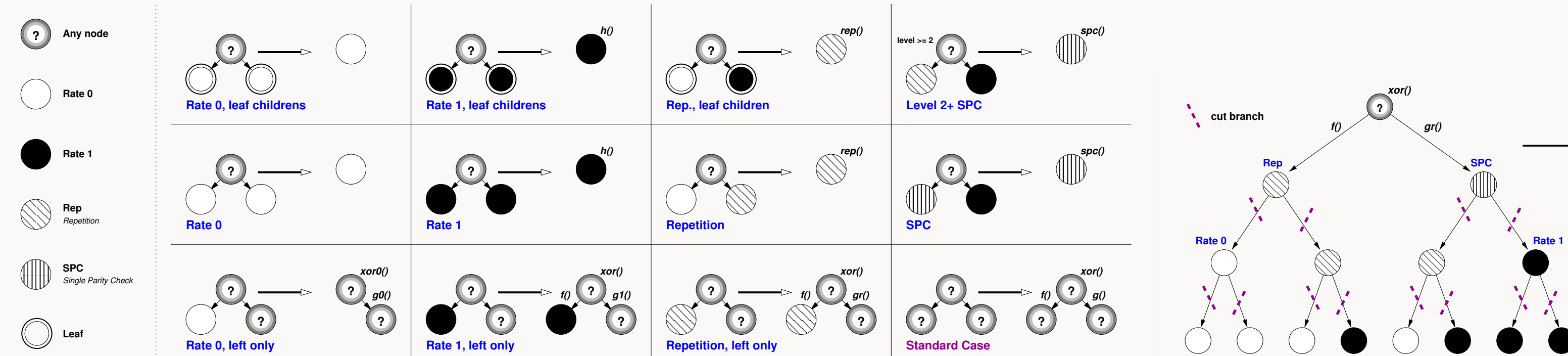


Figure 4: Set of the rewriting rules (left); Example of the rewriting rules application (right)

Binary compression is the key to generate high performance decoders. Fig. 5 presents an example of the P-EDGE sub-tree folding technique used to reduce the binary size of the generated code (frame size $N = 128$).

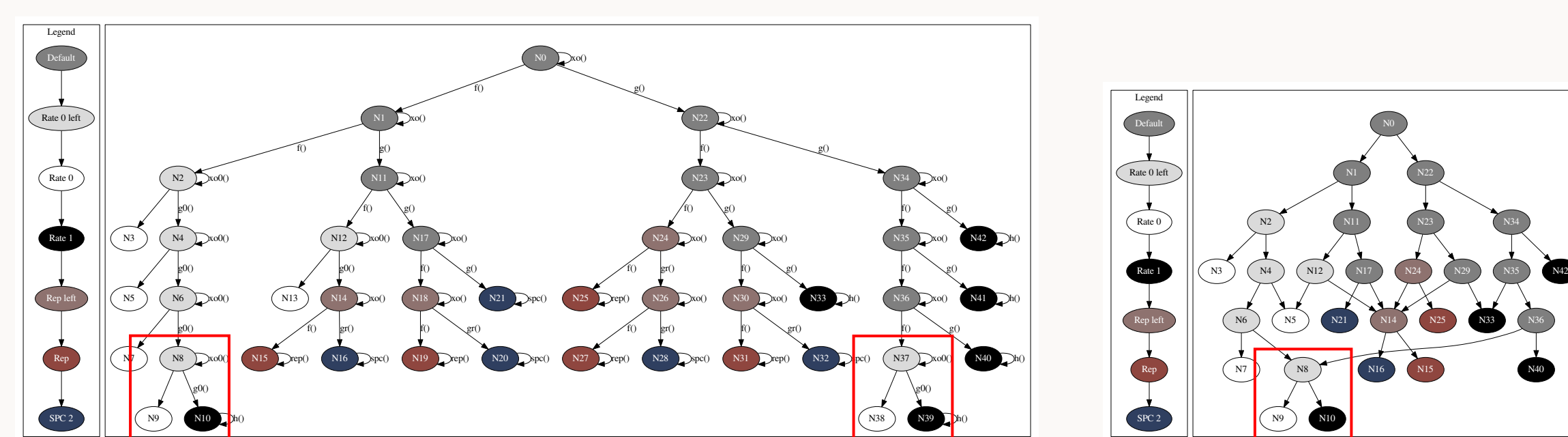


Figure 5: Uncompressed decoding tree (left); Compressed decoding tree (right)

Results

Fig. 2 shows the performance of the P-EDGE generated decoders on various SIMD strategies and on an Intel® Xeon® E31225 CPU @ 3.1Ghz (Sandy Bridge architecture). Higher is better.

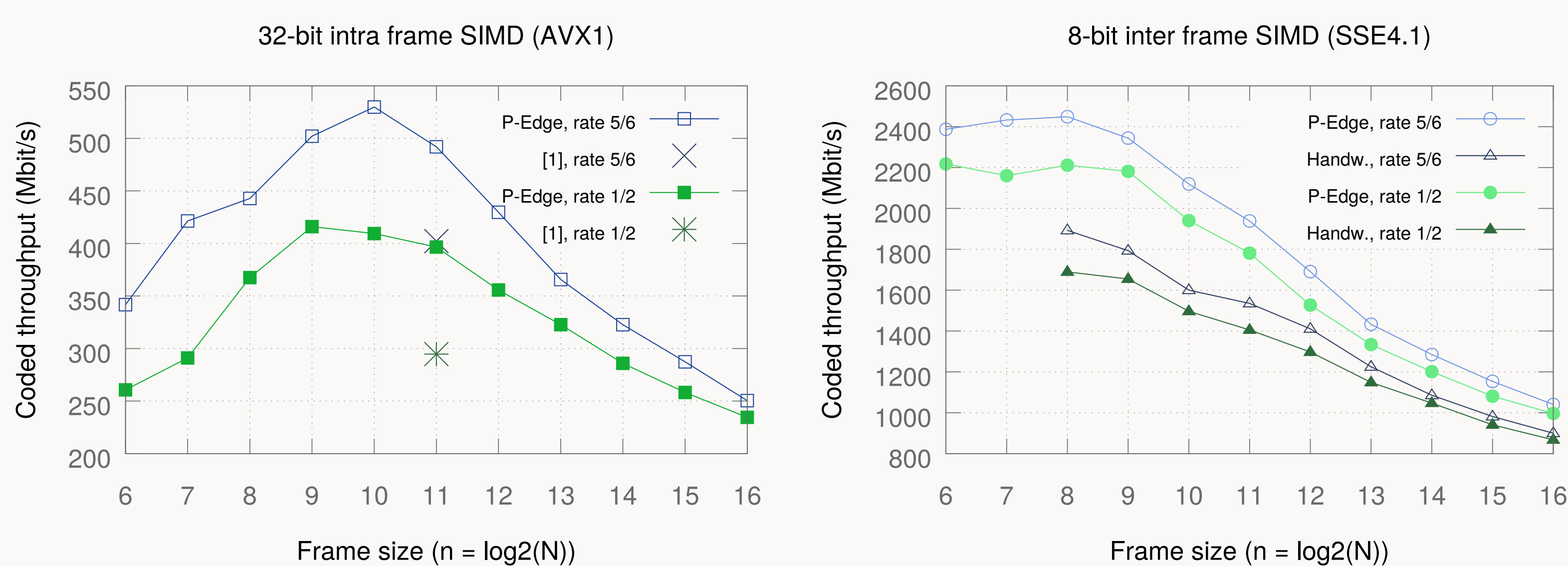


Figure 2: 32-bit floating point intra frame performance comparison: the two cross marks show state-of-the art performance results reported in [1] (left); 8-bit fixed point performance comparison: circles show P-EDGE results and triangles show our former "handwritten" implementation results [2] (right).

The P-EDGE exploration capabilities are demonstrated on Fig. 3: various optimizations can have different impacts on the performance depending on the code rate and the SIMD strategy we use.

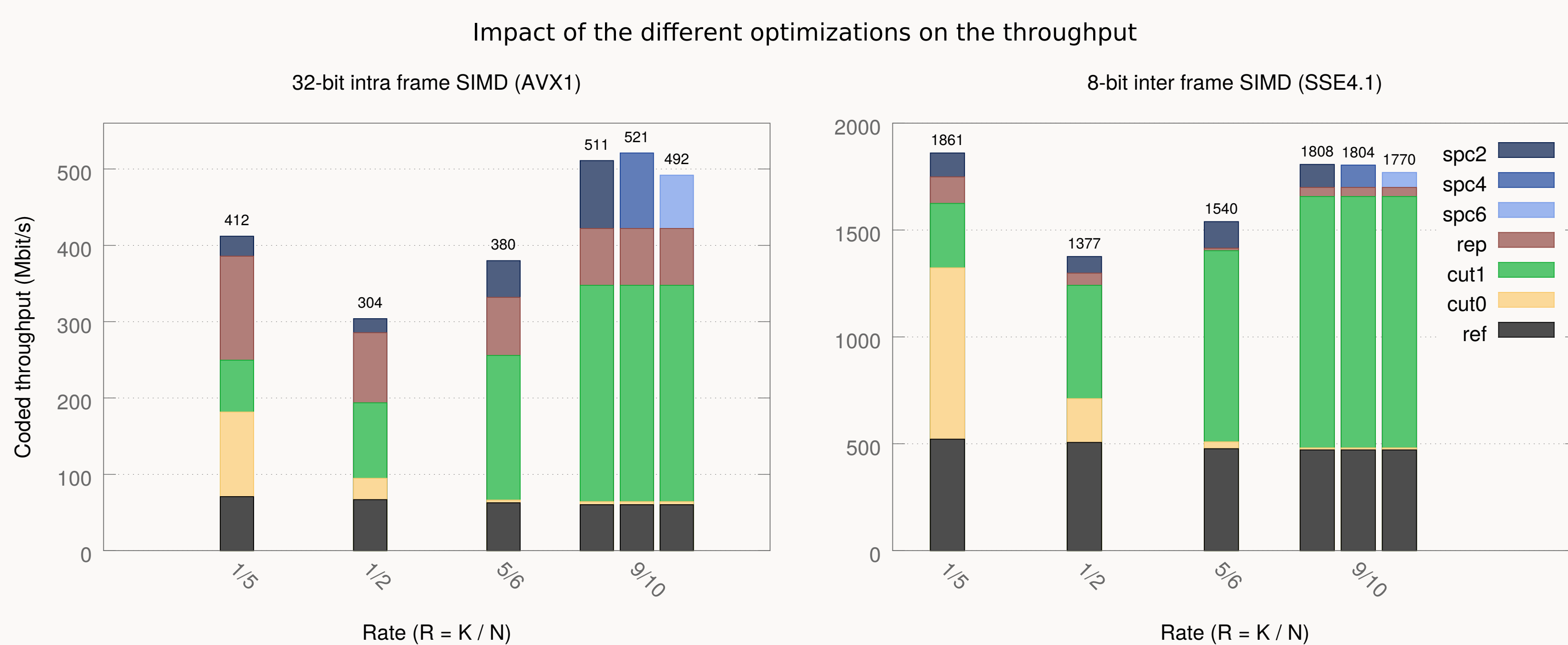


Figure 3: Throughput depending on the different optimizations (frame size $N = 2048$), for intra-frame vectorization on the left and intra-frame vectorization on the right, resp. Compression techniques disabled.

References

- [1] G. Sarkis, P. Giard, C. Thibault, and W.J. Gross. Autogenerating software polar decoders. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 6–10, Dec 2014.
- [2] B. Le Gal, C. Leroux, and C. Jego. Multi-gb/s software decoding of polar codes. *IEEE Transactions on Signal Processing*, 63(2):349–359, Jan 2015.

Genericity and Performance

Clear separation of concerns

- Abstract algorithmic level: ECC experts
- Architecture dependent level: HPC experts

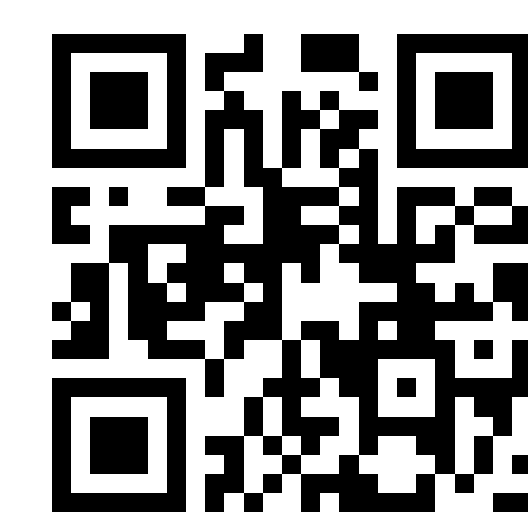
Qualitative and quantitative benefits

- Software design, flexibility
- Performance on par or exceeding state of art

P-Edge reconciles good programming practices and performance!

Contact

Contact e-mail: adrien.cassagne@inria.fr



Acknowledgements

This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the "Investments for the future" Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02).