Two categories of parallelism in Turbo-decoding:

- **Intra frame parallelism**: a single codeword is processed at the time. Computation within the turbo-decoding process are parallelized (trellis transitions, BCJR computations, sub-blocks, ...)
- **Inter frame parallelism**: several frames are processed at the same time. This increases latency but allows more regular memory accesses.

**Contribution**: In this work, we propose a generic and flexible CPU implementation of a turbo decoder that exclusively uses inter-frame parallelism. Experimental results show that our turbo decoder outperforms existing implementations in terms of throughput and energy efficiency.

**The software platform: A Fast Forward Error Correction Tool (AFF3CT)**

http://aff3ct.github.io

- **Support different coding scheme**: Polar, Turbo, Convolutional, Repeat and Accumulate and LDPC (coming soon)
- **Very fast simulations**, take advantage of today CPUs architecture (hundreds of Mb/s on Intel Core i5/7)
  - Written in C++11 (SystemC/TLM support)
  - Monte-Carlo multi-threaded simulations
  - Upto 1000 times faster than MATLAB code
- **Portable**: run on Linux, Mac OS X and Windows
- **Open-source code** (under MIT license)
- For Turbo coding simulations, the following items are configurable: generator polynomial, interleaver, SISO and turbo decoder parameters

**Experiments and Measurements**

- **Conclusion**: In this work a generic and flexible CPU implementation of a turbo decoder that exclusively uses inter-frame parallelism. Experimental results show that our turbo decoder outperforms existing implementations in terms of throughput and energy efficiency.

**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).