Specialized Scalar and SIMD instructions for Error Correction Codes Decoding on RISC-V processors

*Maël Tourres**1,2** , Cyrille Chavet **1** , Bertrand Le Gal **2**, Jérémie Crenne **2,** Phillipe Coussy **1***
***1** Laboratoire Lab-STICC (UMR 6285) Université de Bretagne Sud, France*
***2** Laboratoire IMS (UMR 5218) Bordeaux-INP, France*

The fast deployment of Internet-of-Things (IoT) devices for a few years has been impressive, and the progressive deployment of 5G will accelerate things even further. Indeed, this opens the door to a new generation of standards aiming at a convergence of networks and communication protocols (Wi-Fi, LTE, 4G, etc.). One of the most essential features of these different physical layers, supporting these standards, is the Error Correction process. These codes are designed to improve the reliability of communication links. However, this is also an important performance bottleneck, especially in embedded systems, due to the high computational complexity of the algorithms involved.

Nowadays, such systems need to execute efficiently the error correction decoding algorithms for the most common families of ECC codes such like LPDC & NB-LDPC codes, Turbo-Codes, or Polar codes. The decoding algorithms for these ECC families rely on arithmetic and logic operation sequences, that can be executed in software on a classical CPU design (low cost but low throughput), synthesized in a dedicated ASIC design (high throughput but high cost), or implemented in more flexible hardware designs for in order to mitigate the previous drawbacks (FPGA or ASIP).

In our work, we target an ASIP like design by proposing ECC oriented Instruction Set kits, to improve both throughput and latency, of software-based implementation for ECC decoders on RISC-V cores. Our approach defines sets of custom scalar and vectorized instructions kits adapted to the different families of codes were identified (LDPC, NB-LDPC, Turbo-Codes and Polar codes). The associated architectural instructions of our ECC ISA kits have been designed and integrated the ALU of the RISC-V.

In this context, the work presented in the poster will summarize the designed instruction sets associated with their respective ECC decoding algorithm. These instruction sets were evaluated in an IoT context with low-complexity RISC-V cores (*PicoRV32, IBEX, SCR1, RISCY*). Our experimental results demonstrate a reduction of the required processing clock cycles up to 47.7% for Polar codes, 39.5% for LDPC codes, 16.5% for NB-LDPC codes and 9.7% for Turbo-Codes (LTE) codes with a classical SISD (Single Instruction Single Data) approach.

We also explore SIMD (Single Instruction Multiple Data) instruction extension, providing more impressive speed-ups. The amount of required clock cycles to process coded information blocks is divided by 3, while the throughput is increased also by an order of 3. These values were obtained in simulation and FPGA hardware prototyping. The hardware complexity overhead on FPGA architectures is ranging from 3.3% up to 49.8% and depends on both the modified RISC-V core, and the targeted ECC family.

A set of information provided in this abstract is partially published in **1** whereas the rest is presented in an article that is under review.

1, M. Tourres, C. Chavet, B. Le Gal, J. Crenne and P. Coussy, "Extended RISC-V hardware architecture for future digital communication systems," 2021 IEEE 4th 5G World Forum (5GWF), 2021, pp. 224-229