

# The Variable Precision Processor VXP: preliminary performance results on generic Krylov-based solvers

Yves Durand<sup>1</sup>, Jérôme Fereyre<sup>1</sup>, Cesar Fuguet<sup>1</sup>

<sup>1</sup> Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France.

## Extended abstract

The VXP is a hardware accelerator designed at CEA-List and designed to support extended precision for the resolution of large, ill-conditioned scientific computing problems [1]. Its primary target is to accelerate the convergence of Krylov subspace solvers and eigensolvers, which are the dominant linear kernels of modern applications in computational Fluidics, structure calculation, etc.

The VXP is specifically tailored for accurate floating-point computation using arbitrary length fractional parts (up to 512 bits). It supports IEEE 754 extendable format in memory with byte-aligned data format to optimize memory usage and computing efficiency. The VXP core is built upon a 64 bits RISC V processor which has been modified to feature an extended and variable precision Floating-Point Unit (FPU), a custom Load-and-Store Unit (LSU) and internal registers for up to 512 bits of mantissa and 18 bits of exponent floating-point working registers.

The VXP has been implemented on Silicon as part of EPI system [2]. Nevertheless, since the silicon not yet available (it is expected for Fall'23), this preliminary study has been realized with the prototype versions based on 1/ FPGA Xilinx Virtex UltraScale+ board and 2/ hardware emulation.

The runtime environment is the Real-Time Executive for Multiprocessor Systems (RTEMS rev 5.1). The classical BLAS functions such as gemv, dot, axpy are implemented for extended precision using calls to assembly. VXP's are meant for running in parallel, but this has not been considered in this study: all results are obtained with serial implementation involving only one core. The experiments reported here are coded in C/C++ with extended types. Compilation currently uses a modified version of gcc.

The present work evaluates the performance impact of extended precision 1/ on generic vector-vector, matrix-vector and matrix-matrix operations, 2/ on commonly used Krylov-subspace solvers, namely Jacobi preconditioned Conjugate Gradient (PCG) and Bi-Conjugate Gradient (BiCG) and 3/ on the "periodic reorthogonalization" variant of Lanczos tridiagonalization, which is used for eigen-decomposition. The study involves both dense and sparse matrices. For sparse matrices, we select a subset the Florida sparse Matrix Collection [3] [4], whose actual diagonal size range between 4K and 20K, in order to comply with the limits of our prototyping platform. Otherwise, we use the "ransvd" method [5] to generate pseudo-random dense matrices.

From this set of matrices, we perform two series of evaluations:

1. Iteration count: Using the same library on the same VXP platform, we compare the convergence, i.e. the number of iterations necessary for reaching a predefined error tolerance, between executions in different precisions (including standard double format). This metric is valid for linear solvers, eg PCG and BiCG. For tridiagonalization, we instead measure the number of necessary reorthogonalizations.
2. Cycle count: We measure the execution times of the linear solvers, in clock cycles, using our complete HW/SW, and compared them to “equivalent” kernels running with the same precision on a MPFR [6] software emulation on the same platform.

We observe up to an 8X improvements on kernel iteration count, and up to a 40 % improvement on latency. The count is predictable for dense matrices, and varies with structure for sparse matrices. Nevertheless, the main benefit is the stability gained with the precision. It makes it possible to resolve larger and ill-conditioned systems without costly compensating techniques.

## References

- [1] Y. Durand, E. Guthmuller, C. Fuguet, J. Fereyre, A. Bocco et R. Alidori, «Accelerating Variants of the Conjugate Gradient with the Variable Precision Processor,» chez *29th Symposium on Computer Arithmetic (ARITH)*, Online, 2022.
- [2] E. Consortium, «European processor initiative,» 2022. [En ligne]. Available: <https://www.european-processor-initiative.eu/>.
- [3] T. Davis et Y. Hu, «The University of Florida Sparse Matrix Collection,» *ACM Transactions on Mathematical Software*, 2011.
- [4] University of Texas, «SuiteSparse Matrix Collection,» 2011. [En ligne]. Available: <https://sparse.tamu.edu/>.
- [5] N. J. Higham, *Accuracy and Stability of Numerical Algorithms* (2nd Ed.), Society for Industrial and Applied Mathematics, 2002.
- [6] "MPFR TEAM", «THE MPFR LIBRARY: ALGORITHMS AND PROOFS,» [En ligne]. Available: <https://www.mpfr.org/algorithms.pdf>.