European Processor Initiative

VRP/VxP: VaRiable eXtended Precision RISC-V Accelerator of EPI

DE LA RECHERCHE À L'INDUSTRIE

May 04, 2022

César Fuguet-Tortolero

AGENDA

1. **Context and motivation**
2. **VRP/VXP: VaRiable eXtended Precision RISC-V Accelerator**
   - Hardware view
   - Software view
3. **Hardware Prototypes**
4. **Conclusions**

Context

- **Applications in many scientific domains extensively use linear algebra kernels (e.g. linear solvers or eigensolvers).**
- **The continuously growing complexity in problems led researchers to use Krylov subspace projective methods (instead of direct methods) because of their lower and scalable memory requirements, O(N) instead of $O(N^2)$.**
- **Unfortunately, these methods suffer from important instability, due to accumulated round-off errors, that may prevent solvers to converge.**
- **Current solutions to this issue:**

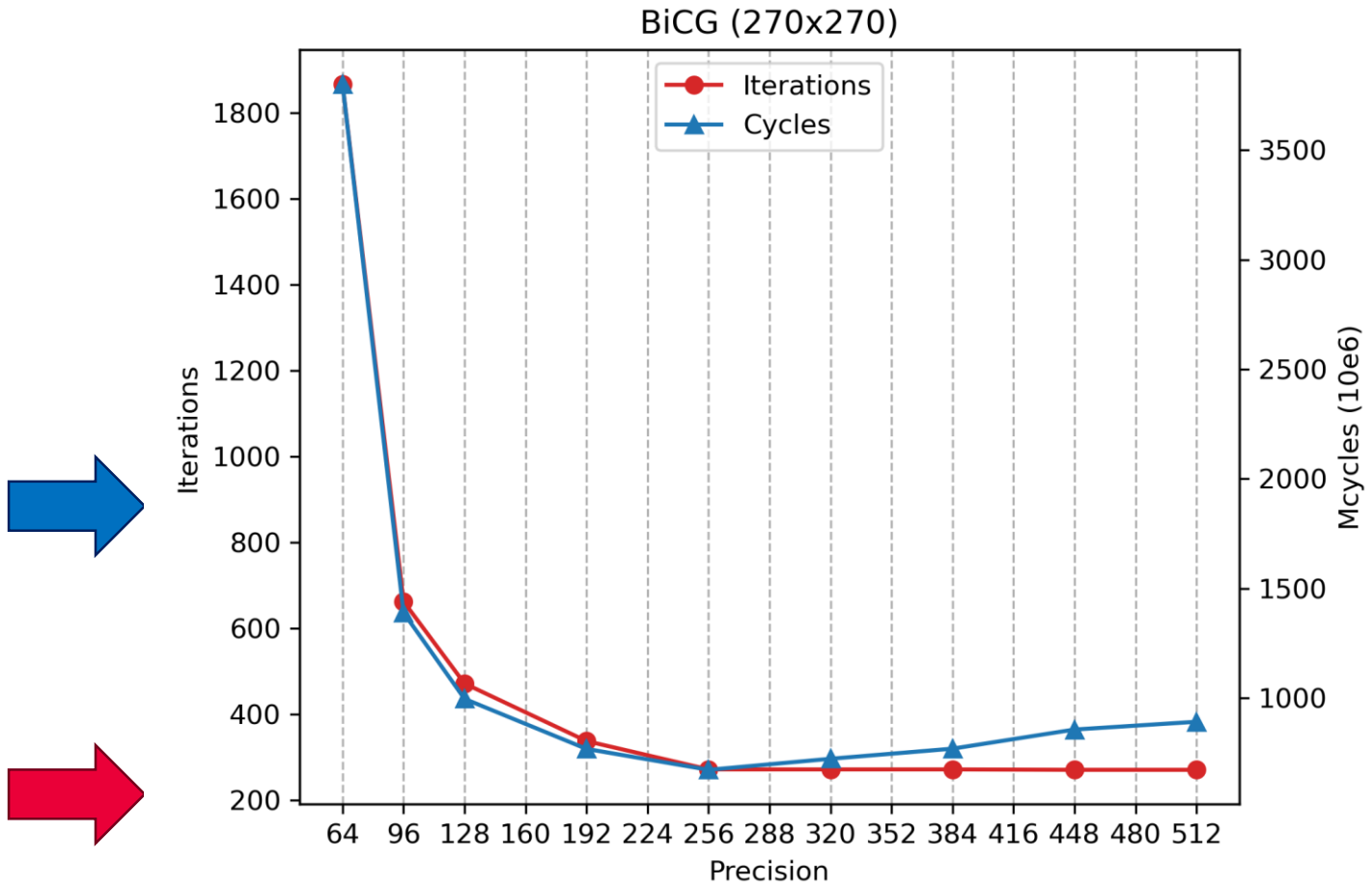
1. **Preconditioning, or re-orthogonalization**
   - Use standard double precision (64 bits).
   - This may be inapplicable due to memory cost and computational complexity.
2. **Extended precision (more than 64 bits)**
   - Increases stability by limiting round-off errors
   - Currently implemented by **very slow** software libraries (e.g. MPFR, libquadmath)

Our solution is a hardware RISC-V accelerator enabling computations with Variable and eXtended Precision Floating-Point (FP) numbers.

Why another hardware accelerator ?
Native hardware support for arithmetic and memory operations enables much higher performance than software-based approaches **(up to x835 speedup)**

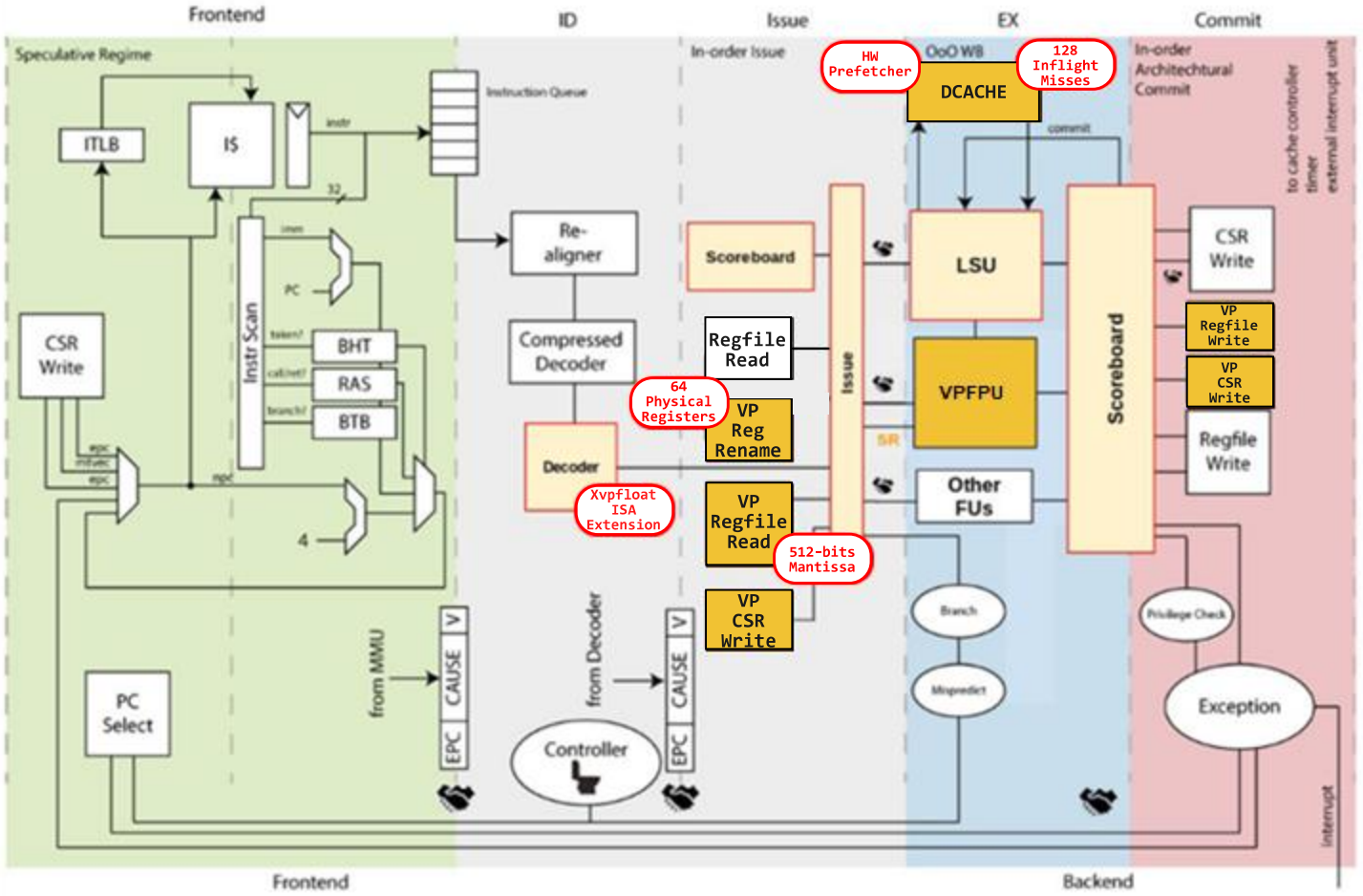
Why variable precision (VP) ?
Allows to tailor the data format to the needs of the application. This reduces both latency and memory footprint.

Why extended precision ?
Allows the solver to converge faster (it reduces the number of iterations)



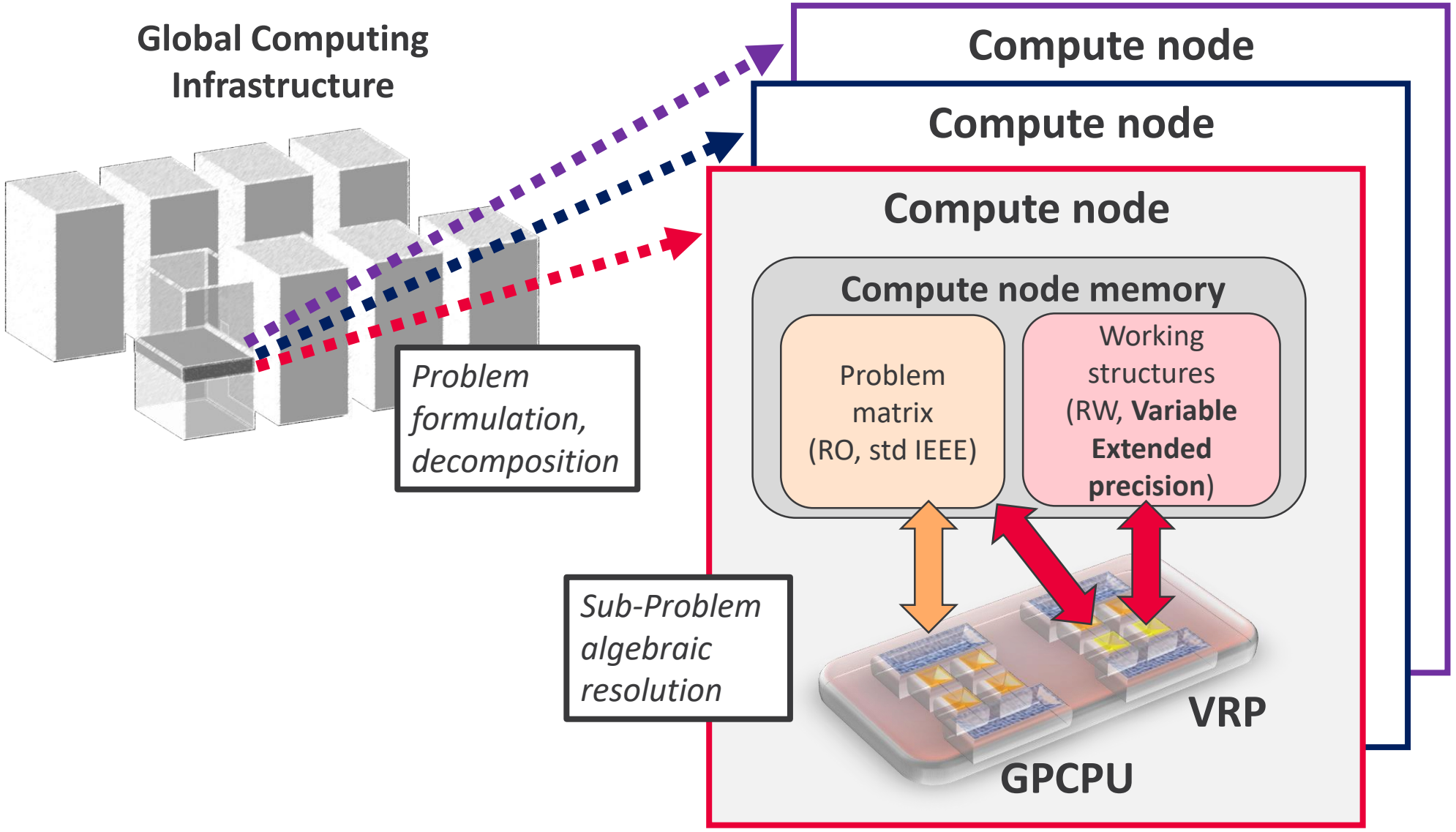
VaRiable eXtended Precision Processor Features (Hardware View)

- **Extended RISC-V core (based on Ariane/CVA6 implementation from ETH Zürich/Univ. Bologna)**
  - Support of linear algebra kernels with complex control structures and complex addressing patterns.

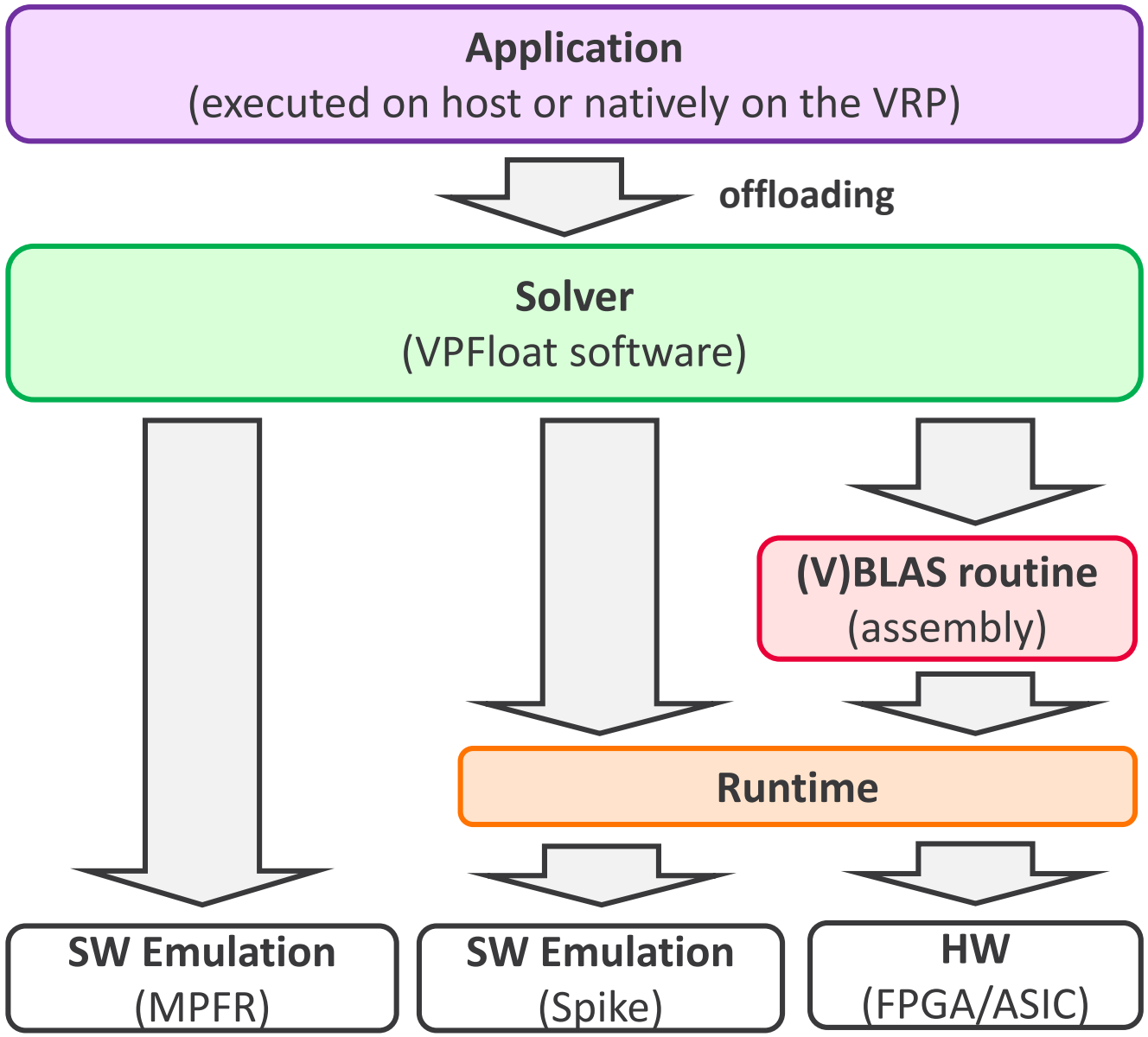


- **Custom ISA extension (namely Xvpfloat) for supporting VP arithmetic, logic, and memory operations on FP numbers.**
  - Dedicated hardware VP FPU
  - Supports up to 512 bits, and 18 bits of significand and exponent size, respectively.
  - Dedicated register file (32 registers) for VP FP numbers.
  - Memory data format complies to the IEEE extendable one (IEEE 754-20008).
- **Disassociate data format and operation**
  - Same code with different precisions
  - Data format is specified by dedicated “environment” registers
- **Advanced L1 data-cache**
  - Byte-aligned transactions to memory.
  - Programmable Hardware prefetch mechanism
  - « Hit under multiple miss » to hide memory latency (up to 128 inflight miss requests).

VaRiable eXtended Precision Processor (Programming model)



VaRiable eXtended Precision Processor (Software Stack)

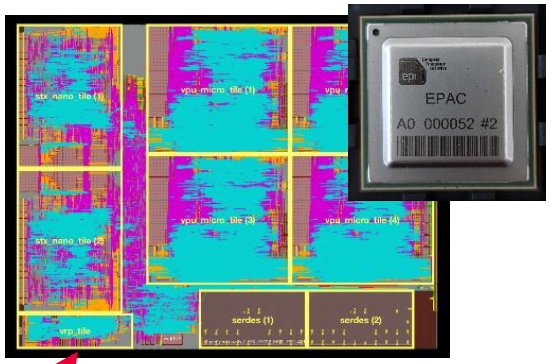



```
VPFloatArray X(EXP_SZ, FRAC_SZ, Ndiag);
…
Nbiter = cg_vp(precision, Ndiag, X, A, B, tol);
```

```
// A*x = b
int cg_vp(int precision, int Ndiag,
          VPFloatArray & x, double *A,
          VPFloatArray b, double tolerance) {
    …
    while (relerror < tolerance) {
        …
        VBLAS::vgemv(precision, n, n, A, p_k, Ap_k, …)
```

```
// y = A*x
void VBLAS::vgemv(int precision, …){
  pser_ec(precision, EC0);          // compute precision
  pser_evp(X.es(), X.fs(), EVP1); // memory precision
  for (int i = 0; i < m; i++) {
    pcvt_d_p(P24, 0);               // acc = 0
    for (int j = 0; j < n; j++) {
      ple(P0, a_ptr, 0, EVP0); // A(m)(n)
      ple(P1, x_ptr, 0, EVP1); // x(n)
      pmul(P0, P0, P1, EC0);   // A(m)(n)*x(n)
      padd(P24, P24, P0, EC0); // acc += A(m)(n)*x(n)
```

Hardware Prototypes

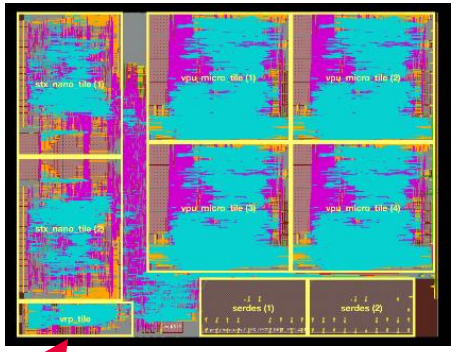


VRP core

EPACTC 1.0
GF22FDX
Q2 2021

VRP core (x2)

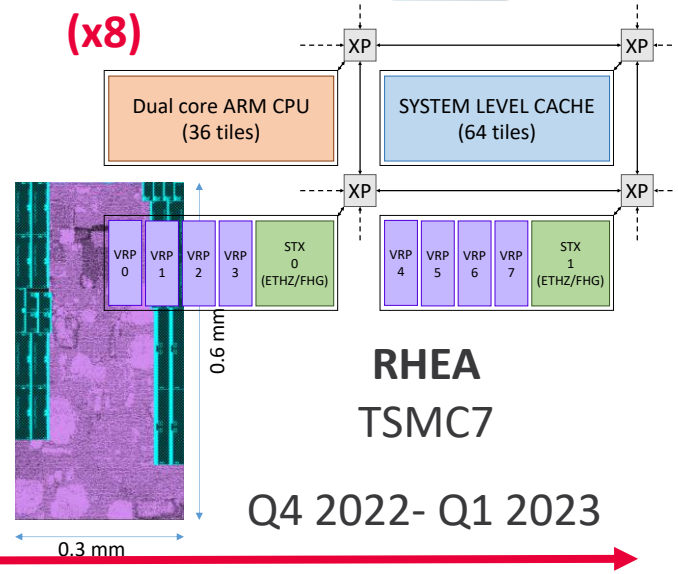
Dual-core VRP tile
FPGA
Q3 2021



VRP core

EPACTC 1.5
GF22FDX
Q3 2022

VRP core (x8)



RHEA
TSMC7
Q4 2022- Q1 2023

EPACTC 1.0 (VRP features)
Up to 256 bits of significand
UNUM Type I memory format
2 dynamic data formats
Clock frequency : 1 GHz

EPACTC 1.5 (VRP features)
Up to 512 bits of significand
IEEE extendable memory format
8 dynamic data formats
Indexed load/store operations
High-Throughput memory subsystem
Clock frequency: 1.2 GHz

RHEA (VRP features)
Same features that in EPACTC 1.5
Two tiles with 4 VRP cores each
Clock frequency: 1.4 GHz

FPGA prototype
2 VRP cores
Clock frequency: 83 MHz

CONCLUSIONS / PERSPECTIVES

- **Software**
  - Consolidating and validating variable-precision solvers and libraries
  - Working on parallel VBLAS routines and sparse matrices support
  - Open position in our laboratory for developping a LLVM-based C compiler with DSL support for the variable precision accelerator (**postulate !** ☺)

- **Hardware**
  - Working on the next generation of the VRP with focus on improving performance of solvers with sparse input matrices
    - Pipeline with Out-of-Order issue and execution.
    - Optimized memory subsystem (L1 cache and hardware prefetcher) for sparse accesses.

ACKNOWLEDGEMENTS

- The work presented is the result of the collaborative work between the members in the LSTA laboratory in the DSCIN division of the CEA List institute.
- Research and software development for rev. 1.0 (UNUM version, incl. compiler) supported by French ANR project **IMPRENUM** leaded by INSA Lyon, with contribution of INPG and CEA.
- EPAC 1.0 and 1.5, and RHEA Implementations supported by **EPI (European Processor Initiative) SGA1 and SGA2.**

Thank you

DE LA RECHERCHE À L'INDUSTRIE

PLANCHE DE PICTOGRAMMES

0 1 2 4 3 5 6 7 8 9