ApproxRISC: An Approximate Computing Infrastructure For RISC-V

Tiago Trevisan Jost, Geneviève Ndour, Damien Couroussé, Christian Fabre, Anca Molnos

ApproxRISC: An Approximate Computing Infrastructure For RISC-V

Authors:

Tiago Trevisan Jost (tiago.trevisanjost@cea.fr), Geneviève Ndour, Damien Couroussé, Anca Molnos,

Speaker:

Christian Fabre

Introduction

- Motivation
  - Many applications are tolerant to a degree of error without a significant loss in quality of results.
  - **Approximate Computing (AC)** aims at exploring the error resiliency of applications.
  - **Lack of support** for **Approximate Computing** in RISC processors
- Challenge
  - Help the rapid investigation of **AC** in RISC-V for **energy reduction** purposes
- Proposal
  - **ApproxRISC**, an infrastructure for approximate computing in RISC-V processors.
    - ISA Extension, Simulator, Compilation

ISA

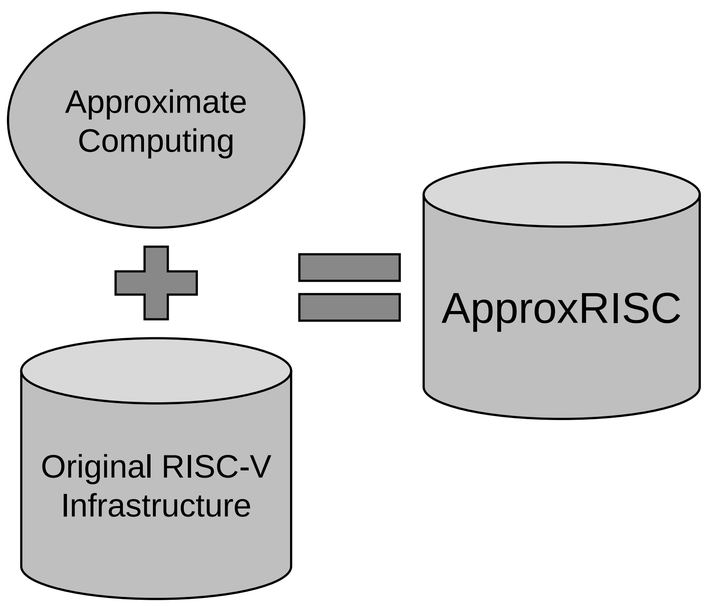
- ISA Extension
  - Set of integer-type instructions for approximate operations.
- Multiple level of accuracy
  - Processor-wide Approximate Bit Width (ABW) value for all AC instructions.

Instruction	Operation Mode
`approx.set.abw`	Set the internal ABW
`approx.get.abw`	Get the internal ABW
`approx.add/addi`	Approximate Additions
`approx.sub/subi`	Approximate Subtractions
`approx.mul/muli`	Approximate Multiplications
`approx.udiv/udivi`	Approximate Unsigned Divisions
`approx.sdiv/sdivi`	Approximate Signed Divisions

ApproxRISC Software

Modifications to the Original RISC-V Infrastructure

Spike ISA Simulator
GNU Binutils
LLVM 3.9



#Pragmas extensions to LLVM

- `#pragma approx`
  - Only AC integers are supported
- `#pragma full_approx`
  - Floating point AC is also supported by means of fixed-point
- Pragma parameters:
  - List of AC variables – explicit & implicit
  - Approximate bit widths.
  - Total width, vs fractional width

```
#define ABW 7
#define WIDTH 32
#define FRAC 10

float Example_full_approx() {
  float a = 44.23, b = 2300.12, d, e = 1000;
  #pragma full_approx(a, b) ABW (WIDTH, FRAC)
True
  {
     d = a + b + e + 1;
  }
  return e;
}
```

Spike

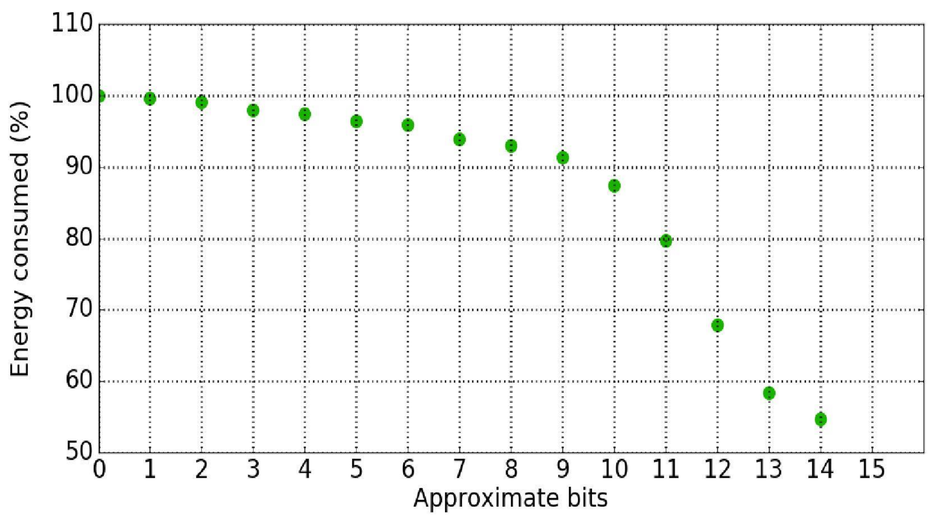
New Features

- Easy-to-use timing capability
- Energy consumption capability
- Instruction categorization
- Region-specific statistics

```
int Example_approx() {
   int a = 44, b = 2300, d, e = 1000;
   #pragma approx(a, b) ABW True
   {
      d = a + b + e + 1;
   }
   return d;
}
```

Experimental Results

- Validation
  - **jmeint** from AxBench [1]
  - Pragmas in the code
  - ABW from 0 (full precision) to 14
  - Energy model according to [2]
  - Up to 50% of energy reduction



Conclusion

- ApproxRISC
  - An infrastructure for rapid investigation of approximate computing in RISC-V processors.
- Future Works
  - Conduct thorough experimentation to show the benefit of AC and our infra
  - Exploration of new instructions

References

[1] A. Yazdanbakhsh et al. “AxBench: A Multiplatform Benchmark Suite for Approximate Computing”. In: IEEE Design Test

[2] D. J. Pagliari et al. “A methodology for the design of dynamic accuracy operators by runtime back bias”. In: DATE, 2017.

[3] RISC-V Foundation — Instruction Set Architecture (ISA). URL: https://riscv.org.

Acknowledgements

This work was partially supported by the French Agence Nationale de la Recherche (ANR), under Grant Agreement, ANR-15-CE25-0015, Project ARTEFACT