Executing ARMv8 Loop Traces on Reconfigurable Accelerator via Binary Translation Framework

Nuno Paulino (nuno.m.paulino@inesctec.pt), João Canas Ferreira (jcf@fe.up.pt),
João Bispo (joao.bispo@inesctec.pt), João M.P. Cardoso (jmpc@fe.up.pt)
INESC TEC & University of Porto, Porto, PORTUGAL

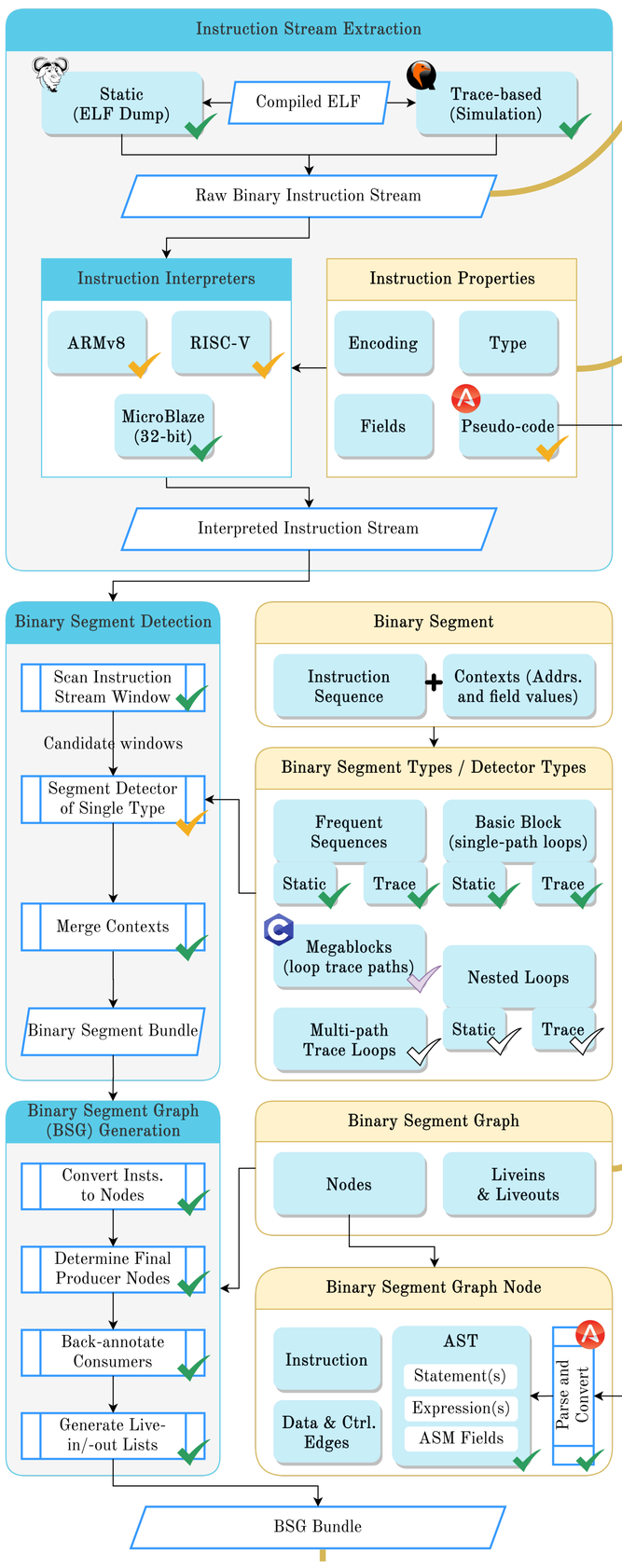
Introduction

Heterogeneous systems allow for tailored solutions to many applications. However, design time is high.

The goal of this project is to devise techniques that map computations onto generated accelerators, aiming for modest but ubiquitous autonomous acceleration in embedded systems.

The purpose of the Binary Translation Stack is to implement this flow.

Binary Translation Stack, Part 1: Detection and Graph Generation



Details, Features & Challenges

Obtaining Instruction Streams: Static vs Runtime

Dumps of the ELF can be easily obtained, but do not express real workload. Real time instruction streams **expose real acceleration opportunities**, but real on-chip retrieval of an instruction trace is greatly dependent on specific system and processor architecture.

We use **QEMU** emulation to explore **opportunities for future self-adaptive heterogeneous systems.**

Future challenge: on-chip real-time retrieval of instruction streams

Defining ISA Properties

Segment detection is independent of the ISA, but late-stage processing requires understanding the instruction set. The BTF contains, **per ISA, a list of properties per instruction.**

This allows **interpreting of ASM fields** and later translations into ASTs and hardware.

Extendable Framework: partial RISC-V support (32iam) added in 2 days

Example Pseudo-code

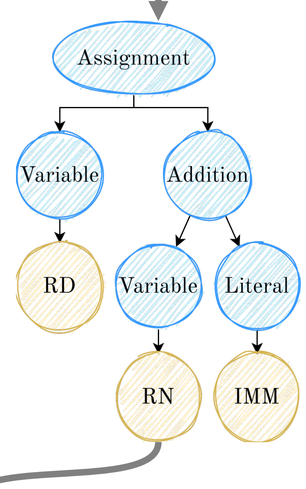
Expresses arithmetic of instructions, using ASM fields as operands.

Example ARMv8 instruction:

0x11000422 is:
`add w2, w1, #0x1`

Pseudo-code *is*
RD = RN + IMM;

Generate parse tree (ANTLR) and transform to Instruction AST



Binary Segments

Repeating instruction sequences with latent ILP and loop pipeling potential.

Example ARM Basic Block

```
lsl xa, xa, #shift
add wb, wb, IMMc
ands sp, xa, xc
b.eq [nzvc], IMMd
```

(Simple) Example Output Code for Single Instruction Unit Generator

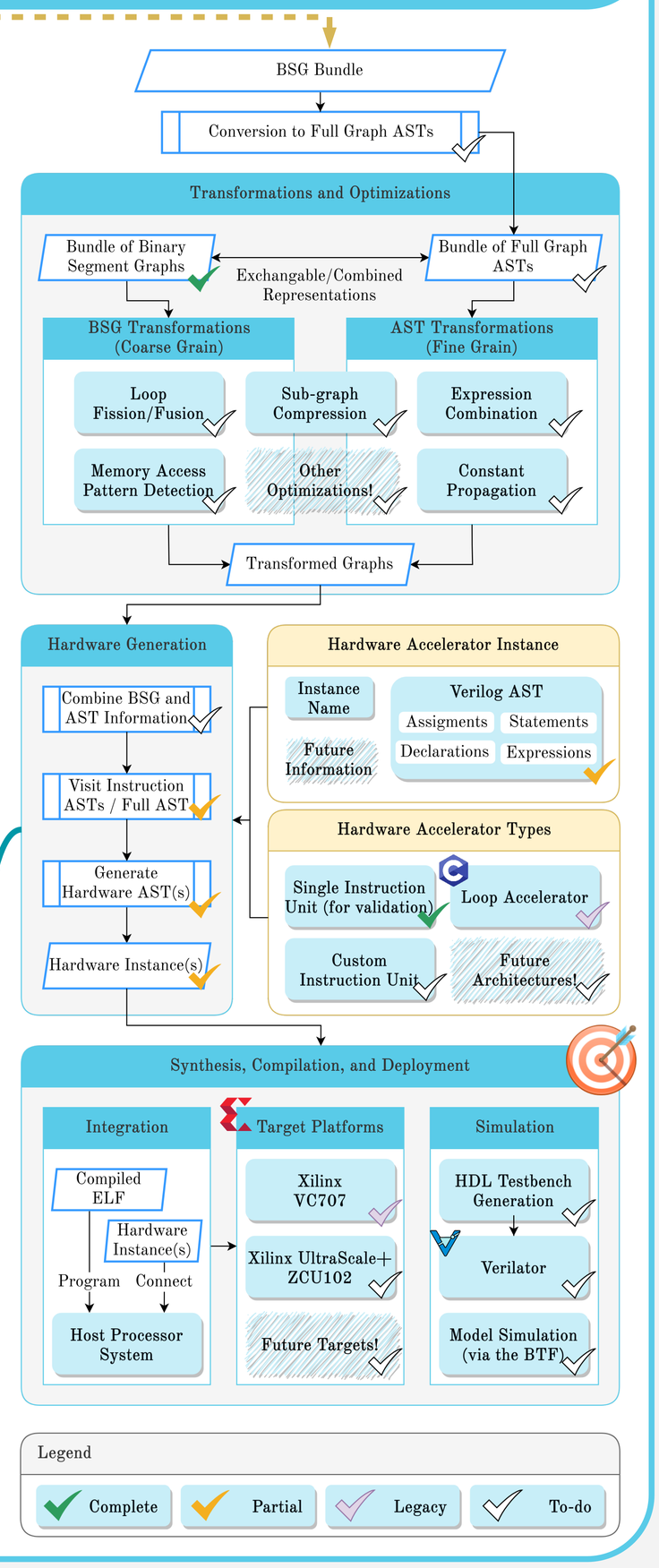
```
/* (...) generation info (...) */
module add_11000422;
   output [31 : 0] w2;
   input [31 : 0] w1;

   // implementation for instruction:
   // add w2, w1, #0x1
   assign w2 = w1 + 32'd1;
endmodule
```

Future challenges: test-bench generation, more architectures, and automated integration

Find the repository on GitHub!
https://github.com/specs-feup/specs-hw

Binary Translation Stack, Part 2: Transforms and Hardware Generation



Results and On-Going Work

Results

- Estimated ILP potential in ARMv8 applications of 4.8 Instructions per Clock Cycle in Trace Basic Blocks
- ISA-independent detection of Binary Segments
- Preliminary automated generation of Verilog from Binary Segments

On-Going Work

- Enhancing the Instruction AST + BSG intermediate representations for further transformations
- Generation and synthesis of loop accelerators for all current ISAs
- Integration into viable host systems an deployment

1. N. Paulino, J. C. Ferreira and J. M. P. Cardoso, "Generation of Customized Accelerators for Loop Pipelining of Binary Instruction Traces", in IEEE Trans. on Very Large Scale Integration Systems, vol. 25, no. 1, pp. 21-34, Jan. 2017
2. N. Paulino, J. C. Ferreira and J. M. P. Cardoso, "Dynamic Partial Reconfiguration of Customized Single-Row Accelerators", in IEEE Trans. on Very Large Scale Integration Systems, vol. 27, no. 1, pp. 116-125, Jan. 2019
3. N. Paulino, J. C. Ferreira and J. M. P. Cardoso, "Improving Performance and Energy Consumption in Embedded Systems via Binary Acceleration: A Survey", ACM Computing Surveys 53, 1, Article 6 (February 2020), 36 pages
4. N. Paulino, (2020): A Breakdown of Binary Acceleration Approaches and Systems. INESC TEC. (DataPaper). https://doi.org/10.13140/RG.2.2.27223.62886
5. Daniel Granhão, "Transparent control flow transfer between CPU and Intel FPGAs", 2019, Universidade do Porto, Faculdade de Engenharia
6. N. Paulino J. C. Ferreira and João M.P. Cardoso, "A Binary Translation Framework for Automated Hardware Generation", 2020, DATE Exhibition

This work was supported by the PEPCC project, "PTDC/EEI-HAC/30848/2017," financed by Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology).