**Automatic Generation of LLVM Backends from LISA Using Processor Designer**

Jeroen Dobbeltaere, Synopsys, Inc.

### LISA: Language for Instruction Set Architectures

- **Register**: / General Purpose Registers
- **Memory**: / Physical Memory
- **I/O**: / Input/Output pins
- **Register File**: / Processor State
- **Instruction Set Description**: / Instruction-set Description
- **Microarchitecture**: / Microarchitecture
- **Memory Hierarchy**: / Memories and Memories
- **ASIP* Characteristics**: / Customizable pipeline with hazards and multiple delay slots

### Processor Designer

- **C/C++ Compiler**: generates
- **Assembler**: generates
- **Linker**: generates

### LLVM Compiler Generation Flow

- **LISA** contains the description
- **LLVM-IR**: helps with creating annotations
- **IR behavior**: provides overview of patterns, machine instructions, intrinsic functions
- **LLVM optimization passes**: can easily be added

### ASIP* Characteristics:

- Custom compiler takes care of hazards and multiple delay slots
- A load has its result available after 5 cycles
- A branch from 3 to 5 delay slots

### Custom scheduler takes care of hazards and multiple delay slots

- **Extraneous int table();
  int partialAdd(int* table, int a, int b);

**Results in:**

```
...}
```

### More details: www.synopsys.com/pd