JAVA DUST: HOW SMALL CAN EMBEDDED JAVA BE?

James Caska and Martin Schoeberl
muvium and DTU
Java Dust

- Java everywhere
  - Even in small microcontroller
    - 1 KB ROM, 1/2 KB RAM?
- Compete with C/ASM
- No JOP included!
- This is not your next RTSJ platform
What is a small microcontroller?
- 8/16 bit

What about doing a small one from scratch?
- 16-bit accumulator machine
- Minimal on-chip memory for RAM/ROM
- RAM contains up to 256 registers
Leros Target

- As small as PicoBlaze
  - 8 bit controller, 2 cycles per instruction
- Faster than PicoBlaze
- No memory size restrictions
- Utility processor
Leros Pipeline
MUVIUM

- Compiler for small microcontroller
- Original targeting Microchip PIC
  - Retarget for Leros
- Input Java bytecode
- Output assembler
- We use currently 16-bit integer
- No real JDK available
- Threading (not yet)
- More needs to be done
  - PIC version can run JemBench
- Plain Java compiler
- muvium bytecode to assembler
- Leros assembler
- FPGA synthesis
RESULTS

- As small as PicoBlaze
- About 1/10 of JOP
- Less instructions than PIC
- 1/3 cycles as PIC
DEMO TIME
Usability

- Leros is open source (BSD)
  - [https://github.com/schoeberl/leros](https://github.com/schoeberl/leros)
- muvium for Leros part of distribution
- FPGA compilers are freely available
- Not too hard to use the tools
Next Steps

- Virtual peripherals
- Play with many core version
- Explore Java specific instructions
  - With minimal size increase
- Make it more usable
  - Do a nice handbook
Conclusions

- Java shall go everywhere
  - Need a small footprint
- Even worth to do a small processor
  - Less than 300 LCs, 1 on-chip memory
- Java can run in 1-2 KB memory
- Give it a try!