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

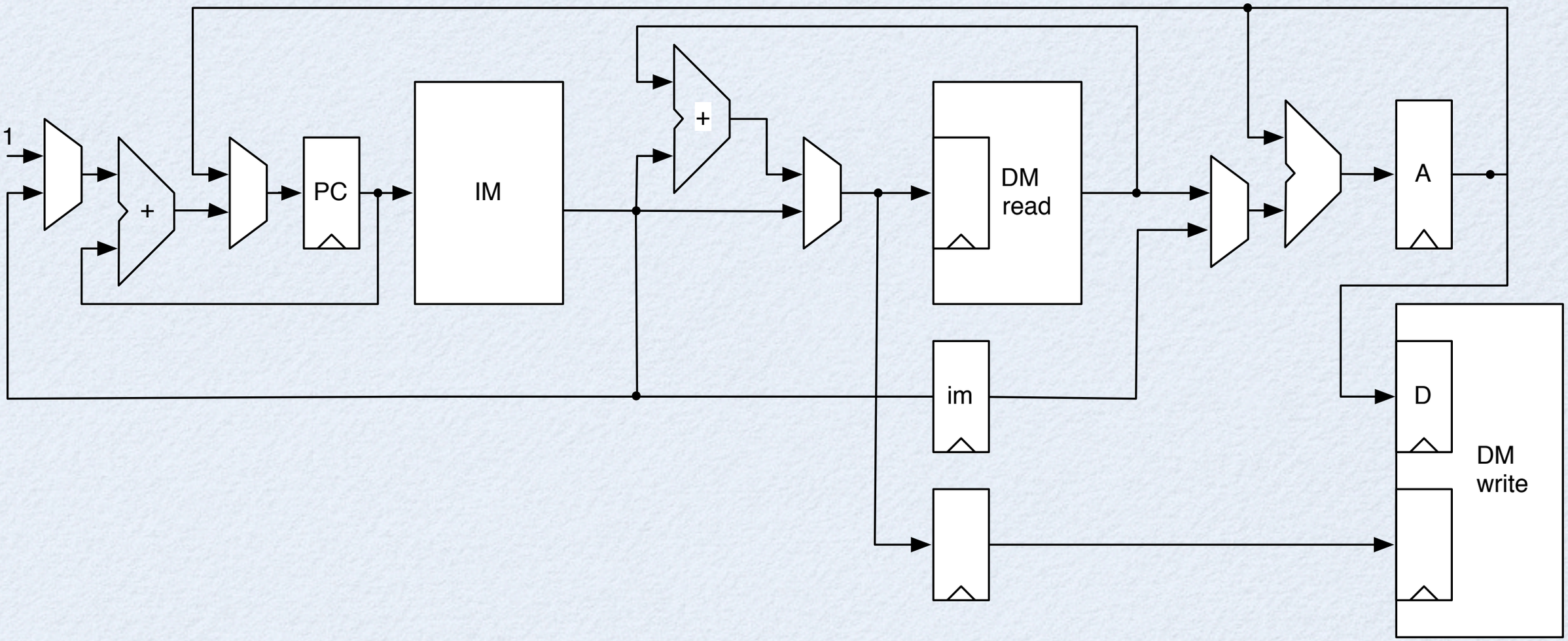
LEROS

- 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

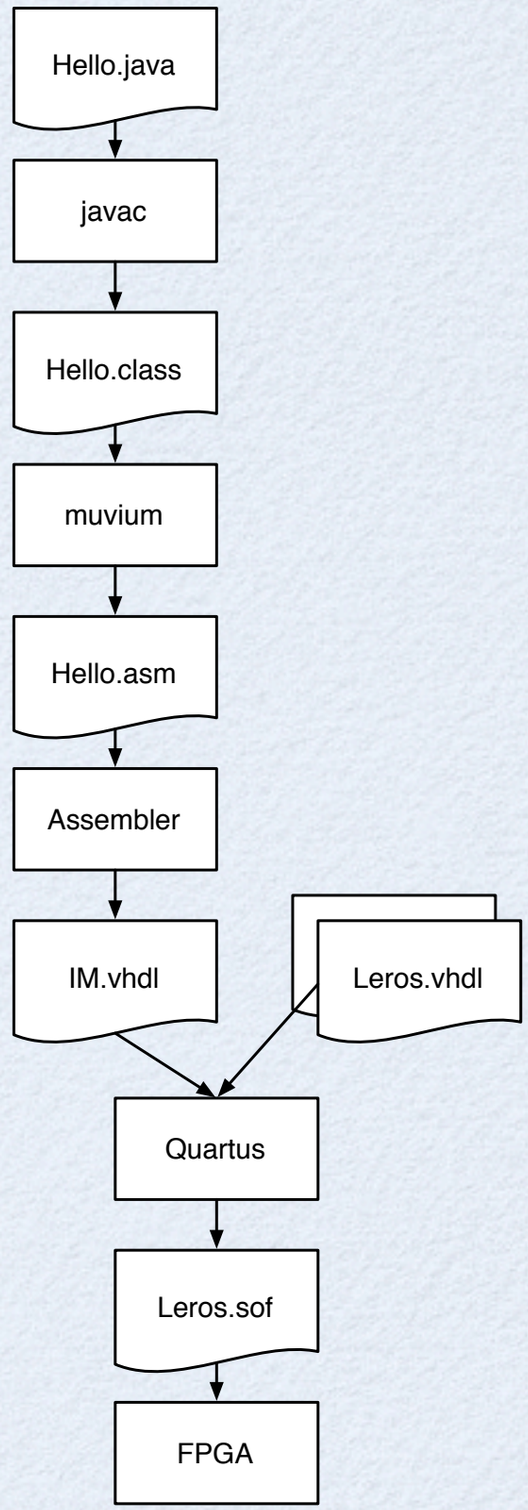
- Compiler for small microcontrollor
- Original targeting Microchip PIC
  - Retarget for Leros
- Input Java bytecode
- Output assembler

Java restrictions

- We use currently 16-bit integer
- No real JDK available
- Threading (not yet)
- More needs to be done
  - PIC version can run JemBench

DESIGN FLOW

- Plain Java compiler
- muvium bytcode 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

Processor	Logic (LC)	Memory (blocks)	Fmax (MHz)
Leros	188	1	115
PicoBlaze	177	1	117
SpartanMC	1271	3	50

Processor	Instructions (byte)	Execution time (clocks)	Cycles/bytecode (clocks)
Leros	574	1577	2.2
Microchip PIC	1326	5196	7.1

DEMO TIME

Usability

- Leros is open source (BSD)
  - 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!