The Delft-Java Engine: An Introduction

John Glossner and Stamatis Vassiliadis
Delft University of Technology
CARDIT - Computer Architecture and Digital Technique
Delft, The Netherlands
{glossner, stamatis}@einstein.et.tudelft.nl
Overview

- Java Properties
- Delft-Java Engine
- Java Hardware Support
- Java Dynamic Instruction Translation
- Link Translation Buffer
- Preliminary Results
- Conclusions
Java Properties

- Object-Oriented Programming Language
- Programmer Supplied Parallelism (Threads)
- Dynamically Linked
- Strongly Typed
  - Statically determinable type state enables simple on-the-fly translation of bytecodes into efficient machine code [Gos95]
- Compiled to Platform Independent Virtual Machine
Delft-J ava Engine

- **RISC-style Architecture**
  - 32-bit Instructions
  - Multiple Register Files

- **Concurrent Multithreaded Organization**
  - Multiple Hdwrr Thread Units
  - Multiple Instruction Issue Per Thread

- **Indirect Register Access**

- **Supervisory Instructions**
  - Branch Java View (bex)

- **Integer & Floating Point**
  - 8, 16, 32, and 64-bit Signed & Unsigned Integers
  - IEEE-754 Floating Point

- **Multimedia Instructions**
  - SIMD Parallelism

- **DSP Arithmetic Extensions**
  - Saturation Logic
  - Rounding Modes

- **32-bit Address Space**
  - Base + Offset + Displacement
Java Hardware Support

- Transparent Extraction of Parallelism
  - Multiple Concurrent Thread Units

- Dynamic Java Instruction Translation
  - Register File Caches Stack With Indirect Access

- JVM Reserved Instruction Used For BEX

- Link Translation Buffer For Dynamic Linking
  - Associates Object Handle And Constant Pool Address With Linked Object Location

- Logical Controller For Non-Supported Translations
  - Thin Interpretive Layer And Java Run-Time
Step 1: Translate Java instr To Indirect Instruction
- `iadd` translates to `add.ind.w32 idx[0] it-1, ix, iy-1`
  - `idx[0]` is an index into the register file

Step 2: Translate Indirect To Direct Instruction
- If `idx[0].rt = 5`, then translated instruction becomes:
  - `add.w32 r5, r4, r4`

In Java-mode, `ix`, `iy`, and `it` are locked together.

In native-mode, they are free to contain independent offsets for Vector operations.
A Global Cache For Dynamically Resolved Names

Associates An Object Handle With A Constant Pool Address.


JVM “invoke” Instructions Maintain High-Level Information Which Allow Use Of A Cache.
Preliminary Results

- 4pt 32-bit Full Complex FFT
  - Compiled with Sun’s javac -O
- Based On C++ Model Of The Delft-Java Processor
- Single-Issue Per Thread, Inorder Processor Model With Adequate Execution Units

![Bar Chart](chart.png)
Conclusion

- **RISC-style Architecture**
  - Multimedia / DSP Datatypes
  - Indirect Register File Access

- **Concurrent Multithreaded Organization**

- **Dynamic Java Translation**

- **Support For Java Dynamic Linking**

- **Transparent Extraction Of Parallelism For Programmers**