The Java Virtual Machine in Hardware — an Exercise in HW/SW Partitioning

Martin Schoeberl
mschoebe@mail.tuwien.ac.at
March 2, 2007

1 Topic

This course is about the implementation of the Java Virtual Machine (JVM) in hardware and the design trade-offs for different HW/SW partitioning schemes.

The students will learn the fundamentals of the JVM and processor architecture. The combination of these two topics will be shown with and practiced on JOP (the Java Optimized Processor). They will learn the design flow starting with building a soft-core CPU in an FPGA up to the Java application that runs on this CPU. Based on the understanding of the design flow the practical aspects of the HW/SW partitioning will be shown.

1.1 Areas Covered

- The Java virtual machine
- Basic processor architecture
- SoC, FPGA design flow
- HW/SW partitioning

1.2 Aim of the Course

The aim of this class is to prepare the students for a career in the emerging area of System-on-Chip with FPGAs and the resulting HW/SW co-design issues.
1.3 EU Cooperation

This class will cooperate with a master class on *Very Small Information Systems* on JOP at the Systems Development Group at the Department of Informatics, Copenhagen Business School (CBS). The students can work on a joint project with students from Copenhagen. This special offer provides a great opportunity to learn distant cooperation. Furthermore, the two classes will exchange their final reports for a peer review.

2 Work by the Students

The work will be done in groups of up to 4 in the laboratory. The students are free to choose exercises that fit to their skills and interests. A report in a short paper format (up to 4 pages) and a short presentation of one project will conclude the course.

2.1 Introductory Exercise: 10 Points

The first exercise is a *Hello World* program running on JOP. This work is supervised and will help the students to get used to the design flow. After this introduction the students are free to select one project.

2.2 Thread Example: 10 Points

The second exercise gives an introduction into programming of periodic real-time threads.

2.3 Project: 60 Points

A project is autonomous work and each project can only be done by one group. A project can also be done in cooperation with students from Denmark at the sister course on JOP.

- Table/lookupswitch in microcode
- Instruction timing of the Cjip
- Eclipse plugin
- FPU SoftFloat comparison
- Bytecode optimization
- synchronized change in JOPizer
- Scoped memory
- EDF scheduler
- java.lang.Thread
2.4 Final paper: 20 Points

The final paper should describe the problems and solutions of one task or project. The paper has to be written in English and will be reviewed by students of a sister course in Copenhagen. The students in Vienna have to review the papers from Copenhagen.

3 Organization

We will meet every week for a talk or supervised lab hours. There will be one introductional exercise at the beginning of the course. The other practical exercises can be selected from a list by the students to cover their interest (e.g., more hardware oriented or more software oriented).

At the end of the course a short paper (up to 4 pages) about the project has to be written. The papers from this course will be exchanged with the papers from the class at CBS for peer review by the students. The corrected paper is then presented in a 10 minutes talk with a live demo of the project.

3.1 Lecture Topics

The tentative list of the talks:

- Java and the JVM
- Design flow: from VHDL and Java sources to a running system
- VHDL crash course (optional)
- Bytecodes, the instruction set of the JVM
- JOP, the Java Optimized Processor in an FPGA
- Design decisions for JOP
- Real-time Java

3.2 Grading

The grade is based on the exercises or project work, contributions to JOP and the final short paper. The grading scheme is shown in Table 1.
<table>
<thead>
<tr>
<th>Points</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>86-100</td>
<td>S1</td>
</tr>
<tr>
<td>70-85</td>
<td>U2</td>
</tr>
<tr>
<td>60-69</td>
<td>B3</td>
</tr>
<tr>
<td>50-59</td>
<td>G4</td>
</tr>
<tr>
<td>0-49</td>
<td>N5</td>
</tr>
</tbody>
</table>