Laboratoire d'Architecture des Processeurs
IN-F Ecublens
CH-1015 Lausanne
http://lapwww.epfl.ch/

Semester project
Winter 2003-2004

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Java Virtual Machine on FPGA based Platforms

Christophe Dubach – Computer Science, 7th semester

Transparent PLD use from Java

Alok Menghrajani – Computer Science, 7th semester

Introduction

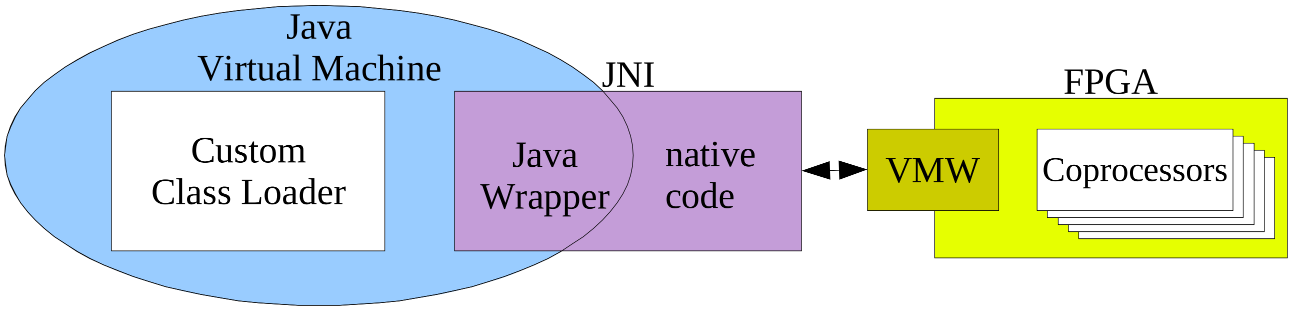
Our projects are based on the RokEPXA board (developed previously at the LAP). This board has a hybrid architecture; it combines a reprogrammable FPGA hardware with a general purpose processor (ARM). By using the virtualization layer developed by Miljan Vuletic (VMW), we will extend the possibilities of the Java Virtual Machine (JVM).

Objectives

– Creation of a complete Linux-based development environment and a JVM.
– Using coprocessors from Java
(Christophe Dubach's part of the project).
– Getting the JVM to use Christophe Dubach's code in a transparent way
(Alok Menghrajani's part of the project).
– Porting IDEA (encryption library) to Java, and using it as a test application.

Summary

We were able to create and document a complete Linux-based environment (Linux 2.6.0, native compiler, cross-compiler, Kaffe 1.1.3 JVM, network support,...).
Using Java Native Interface (JNI), we implemented a library that allows Java code to call the coprocessors. It is also possible to proceed the other way round (call Java methods from the coprocessor). We used a custom class loader which modifies the bytecode (using BCEL library) at runtime. It provides a transparent and portable solution (independent of the JVM) to use the coprocessors without modifying the original source code.
We were able to successfully test IDEA with different configurations (with and without coprocessor) and architectures (x86, ARM).
Our implementation has very little overhead, we can gain a lot of performances in the case of large coprocessors.



Future Work

There are several areas that could extend the possibilities of our system. Such areas include automatic conversion of Java code into VHDL, dynamic (based on runtime statistics) determination of which coprocessors to use and exploring the possibilities of callbacks from the coprocessor.