Accelerators and coarse grained reconfigurable architectures

Energy efficiency is becoming more important with every new generation of electronic device. Either because of cooling issues, such as in datacentres, or in order to achieve a decent battery life.

We will go into the aspects that make a general purpose processor rather inefficient (typically, only about 6 percent of energy is actually used by the ALU) and how the use of accelerators can be used in order to improve the energy efficiency. According to some work the difference in energy efficiency between a properly designed accelerator and a general purpose processor can be as high as a factor 500. Accelerators that achieve such high efficiencies are very specialized and therefore really only suitable for a very specific purpose.

With new standards and algorithms appearing very regularly a less specialized or reconfigurable architecture will offer more flexibility, but at a price in energy efficiency. The most popular reconfigurable devices at the market today are Field Programmable Gate Arrays (FPGA). These allow the implementation of (almost) any digital electronic circuit due to their ability to reconfigure the functionality if the building blocks, Combinatorial Logic Blocks (CLB), and the interconnect network at a gate-level granularity. However, this requires that each CLB has to be configured individually, and that each bit coming from or going to the CLBs has to be routed over the interconnect network. This leads to a large configuration memory (with a high leakage current), longer wires (a long critical path) and a large area.

In order to achieve good energy efficiency for reconfigurable architectures some Coarse Grained Reconfigurable Architectures (CGRA) have been proposed in the past. These architectures do not reconfigure on the gate-level but on the level of a functional unit, such as an adder or a multiplier. This reduces the amount of configuration memory required significantly.

In our own research we have taken this a step further and allow run-time construction of specialized processors. Our architecture consists of functional units that can be connected to instruction decoders to allow construction of VLIW and SIMD (or any combination of these) processors depending on the application requirements.

Bio

Mark Wijtvliet received the M.S. degree in Embedded Systems from Eindhoven University of Technology (TU/e) in 2011. And subsequently worked there until 2013 as a researcher with a focus on Algorithmic Skeleton compilation techniques for Field Programmable Gate Arrays (FPGAs). Currently he is pursuing the Ph.D. degree from the Electronic Systems Group from TU/e in collaboration with IMEC-NL. His research interests include low-energy computer architecture, spatial computing, reconfigurable computing and and near memory computing.

Assignment

In this assignment the students will work with the reconfigurable architecture developed at Eindhoven University of Technology. This architecture is a CGRA that allows you to construct a processor that matches the structure of the application that you want to run on it very closely. This leads to big performance improvements as well as improvements in energy efficiency.

You will get a reference architecture implementation (constructed with our CGRA) that is quite similar in resources to what you find in a typical RISC processor. You will also receive a naive implementation of the algorithm for this processor. Your job is to modify the architecture (which can be done by modifying a configuration file) and the application source code. By tuning one to match the other, you can tune the design to obtain a optimal balance between performance, power and area.