A Computational Intelligence Based Coarse-Grained Reconfigurable Element

C. Hart Poskar, Peter J. Czezowski, and Robert D. McLeod
Department of Electrical & Computer Engineering
University of Manitoba, Winnipeg, Manitoba, R3T 5V6 Canada
{hpos | czezow | mcleod}@ee.umanitoba.ca

Computational intelligence techniques, such as neural networks and fuzzy systems, are increasingly being employed in real-world applications. Accordingly, much design effort is going into developing customizable modules to meet required hardware/software specifications. We have prototyped a fuzzy neuron as a coarse-grained reconfigurable element to be used in such a module. This module contains multiple reconfigurable fuzzy neurons, together with built-in memory, interface and fine grained reconfigurable logic to implement a fuzzy neural network in the fashion of a system-on-a-chip. The result is a dynamically reconfigurable computational intelligence based control/decision making system which features a parallel structure and in-situ learning.

Design issues in the development of a JAVA-processor for small embedded applications

Hagen Ploog, Tino Rachui, Dirk Timmermann
Department of Electrical Engineering and Information Technology
University of Rostock, Germany
hp@e-technik.uni-rostock.de

This poster presents some design issues in the development of a JAVA-processor according SUN92s JavaCard 2.0 API for use in small embedded applications which could be realized with FPGAs. We employed this API because threads and garbage collection are not defined within this specification which leads to small area requirements. As our current solution is microcode-based we demonstrate that the footprint of the Java-processor can be reduced when using loosely coupled state machines (a microcode-sequencer and three slave state machines). Each slave state machine can HALT the microcode-sequencer while itself is still running. Furthermore we discuss some architecture details on implementing the stack on such systems as Java machine implementations are stack-based computer architectures.

Dynamically Programmable Cache Evaluation and Virtualization

Mouna Nakkar, David G.Bentlage, John Harding, David Schwartz, Paul Franzon, and Thomas Conte
North Carolina State University

Dynamically Programmable Cache (DPC) is a novel architecture for embedded processors which offers high memory bandwidth and fast data accessibility. DPC processors merge reconfigurable arrays with data cache blocks at various cache levels to create multi-level reconfigurable machines. This will provide high memory bandwidth for FPGA cells and higher computation capacity per memory access. In addition, DPC machines implement a multi-context switching (Virtualization) concept. Virtualized DPC machines have two advantages: 1) they allow implementation of large subroutines with fewer FPGA cells, 2) and they can execute several operations in parallel resulting in faster execution time. The speedup improvements for the DPC machine are shown to be 5X faster than an Altera FLEX10K FPGA chip and 2X faster than a Sun Ultra1 SPARC station for three different algorithms (convolution, motion estimation, and run-length coding).