Formal Methods for Engineering Special-Purpose Parallel Systems Introduction to Minitrack

Ali E. Abdallah
Centre for Applied Formal Methods
South Bank University
103 Borough Road
London, SE1 0AA, U.K.
A.Abdallah@sbu.ac.uk

Wayne Luk
Department of Computing
Imperial College
180 Queen's Gate
London SW7 2BZ, U.K.
wl@doc.ic.ac.uk

The design and application of special-purpose parallel systems have witnessed a rapid industrial growth in recent years and have become a key technology for the new information age. Recent technological advances in Field-Programmable Gate Arrays (FPGAs) coupled with Hardware Compilation techniques allow parallel programs to be turned into application-specific circuits rather than machine code running on a standard processor. Due to the massive parallelism in the hardware implementations, programs can potentially achieve enormous improvement in speed. Equally important, the applications can be implemented by programmers systematically applying parallel programming principles and techniques rather than thinking about hardware design and structures. Design methods and tools for effectively enabling this technology have been an exciting research area for the past decade. This Minitrack specifically focuses on the use of Formal Methods for engineering special-purpose parallel systems. The main purpose is to bring together an international audience of researchers and practitioners actively involved in the formulation, design, and development of correct and efficient parallel systems. The Minitrack programme consists of six accepted papers.

The paper *Stream Processing Hardware from Functional Language Specifications* by Simon Frankau and Alan Mycroft decribes the design of a first order functional programming language SASL for compiling*restricted* functional programs directly into hardware. The main design aim is to achieve a statically allocated stream-processing language where input streams cannot be *rewound*.

The paper *Field Modifiable Architecture with FPGAs and its Design/Verification/Debugging Methodologies* by M. Fujita, et al. proposes Field Modifiable Architecture (FMA)that exploit microcontrollers and IP libraries to develop effective harware/software co-design solutions.

The paper *Formal Behavioural Synthesis of Handel-C Parallel Hardware Implementations from Functional Specifications* by A. Abdallah and J. Hawkins emphasises the use of algebraic laws of functional programming to transform high-level specifications into Handel-C programs which, in turn, can be compiled into efficient circuits running on reconfigurable hardware.

The paper *An Approach to the Introduction of Formal Validation in an Asynchronous Circuit Design Flow* by Dominique Borrione et. al. describes a method for asynchronous circuit design which includes automatic synthesis, simulation, and formal verification.

The paper *Towards Verifying Parametrised Hardware Libraries with Relative Placement Information* by Steve Mckeever, Wayne Luk and Arran Derbyshire presents a framework for verifying compilation tools for parametrised hardware libraries. Pass separation techniques are used to verify a procedure for compiling designs with relative placement information into designs with explicit placement coordinate information.

The paper *Combining Imperative and Declarative Hardware Descriptions* by Tim Todman and Wayne Luk describes an approach for hardware development that involves both imperative and declarative descriptions.

We hope that this Minitrack will provide a lively atmosphere for productive discussions on many aspects of the presented papers. We would like to thank all the authors for their submissions and all the referees for their valuable professional reviews. We take this opportunity to particularly thank both the track chair, Hesham Rewini, for encouraging us to organize this Minitrack for the first time at HICSS, and Eileen Robichaud Dennis for all her hard work!

0-7695-1874-5/03 $17.00 © 2002 IEEE