TRIMARAN

An Infrastructure for Research in Instruction-Level Parallelism

http://www.trimaran.org

The Mission

Trimaran is an integrated compilation and performance-monitoring infrastructure. The architecture space that Trimaran covers is characterized by HPL-PD, a parameterized processor architecture supporting novel features such as predication, control and data speculation and compiler controlled management of the memory hierarchy. Trimaran also consists of a full suite of analysis and optimization modules, as well as a graph-based intermediate language. Optimizations and analysis modules can be easily added, deleted or bypassed, thus facilitating compiler optimization research. Similarly, computer architecture research can be conducted by varying the HPL-PD machine via the machine description language HMDES. Trimaran also provides a detailed simulation environment and a flexible performance-monitoring environment that automatically tracks the machine as it is varied.

Structurally the Trimaran system consists of

- A machine description facility for describing ILP architectures.
- A parameterized ILP architecture called HPL-PD.
- A compiler front-end for C. It performs parsing, type checking, and a large suite of high-level (i.e. machine independent) optimizations.
- A compiler back-end parameterized by a machine description, performing instruction scheduling, register allocation, and machine-dependent optimizations. Each stage of the back-end may easily be replaced or modified by a compiler researcher.
- An extensible IR (intermediate program representation) that has both an internal and textual representation, with conversion routines between the two. The textual language is called Rebel. This IR supports modern compiler techniques by representing control flow, data and control dependence, and many other attributes.
- A cycle-level simulator of the HPL-PD architecture, which is configurable by a machine description and provides run-time information on execution time, branch frequencies, and resource utilization. This information can be used for profile-driven optimizations as well as to provide validation of new optimizations.
- An integrated Graphical User Interface (GUI) for configuring and running the Trimaran system. Included in the GUI are tools for the graphical visualization of the program intermediate representation and of the performance results.

Why Trimaran?

The infrastructure is used for designing, implementing, and testing new compilation modules to be incorporated into the back end. These new modules may augment or replace existing modules, and may be the result of research in instruction scheduling, register allocation, program analysis, profile-driven compilation and more. Although there are several compiler infrastructures available to the research community, Trimaran is especially useful for the following reasons:

- It is especially geared for ILP research. It provides a rich compilation framework. The parameterized ILP architecture space allows the user to experiment with machines that vary considerably in the number and kinds of functional units, register files and instruction latencies.

- The machine configurations can be described both using the powerful machine description facility (HMDES) and using the Trimaran GUI.

- The modular nature of the compiler back end and the single intermediate program representation used throughout the compiler back end makes the construction and insertion of new compilation modules into the compiler especially easy.

- The framework is already populated with a large number of existing compilation modules, providing leverage for new compiler research and supporting meaningful experimentation instead of running "toy" programs.

- The Trimaran Graphical Interface makes the configuration and use of the system surprisingly easy.

- There's a commitment on our part to releasing a robust, tested, and documented software system.

- *It is a vehicle for researchers to publish their software to the broader community. This process is managed by the Trimaran Council consisting of Krishna Palem (chair), Vinod Kathail, Wen-Mei Hwu and three nominees from HP Laboratories, The University of Illinois and The Georgia Institute of Technology.*

Current Impact

Community: The Trimaran user community is tracked through surveys and interactions with the support group. Since the release of Trimaran version 2, more than *two thousand downloads* from unique sites have been logged. More than *thirty* research groups worldwide are currently using Trimaran in their research. Some of the Universities that are actively use Trimaran as a part of their research include the National University of Singapore, Technical University of Munich, INRIA, AC-Grenoble, Weizmann Institute, Politecnico di Milano, Ghent University, Appalachian State University, University of Alberta, University of Toronto.

Highlights: The Trimaran user community uses Trimaran in a range of compiler and architecture research areas. Some highlights:

Politecnico di Milano, Department of Electronics and Information
Hardware fault detection for VLIW architectures

- Fault tolerance in the future
- Hardware and software techniques to exploit the redundancy of the architecture and word width
- Papers submitted for publication

Ghent University, Department of Electronics and Information Systems
Instruction Scheduling and Multithreading

- Minimizing the number of false memory dependences for improved scheduling – Master of Science thesis in progress
- Compiler Generated Multithreading to Alleviate Memory Latency

University of Toronto
Compilation for Digital Signal Processors

- Exploiting Dual Memory Banks in DSPs - Master of Science thesis
- TI-C6x code generator module integrated with Trimaran

Pennsylvania State University
Power-Aware Scheduling for VLIW Architectures

University of California, Davis
Superblock Scheduling

University of Texas
Hyperblock Formations

Scholarly Accomplishments

Numerous papers, theses, class work and software have benefited from the Trimaran infrastructure. Some highlights:

Papers

Lori Carter, Beth Simon, Brad Calder, Larry Carter and Jeanne Ferrante, *"Predicated Static Single Assignment"* Proceedings of the International Conference on Parallel architectures and Compilation Techniques, October 1999.

Kristof E. Beyls**,** Erik H. D'Hollander, *"Compiler Generated Multithreading to Alleviate Memory Latency"* Journal of Universal Computer Science, Special Issue On Multithreading, October 2000.

Glenn Altemose, Cindy Norris, *"Register Pressure Responsive Software Pipelining"* Proceedings of the 16th Symposium on Applied Computing, March 2001.

Courses

Technische Universität München
Computer Architecture and Organizations
Microprocessor and Microprocessor Systems

Politecnico di Milano
Computer Systems and Introduction to VLIW Architectures

Georgia Institute of Technology
EPIC Architecture
Compiler Design

Software

Pressure Responsive Pipelining - Appalachian State University

- Cooperative register allocation and software pipelining optimization integrated with Trimaran.

Tritanium - National University of Singapore and George Washington University

- Itanium-like parametric processor configuration and Itanium code generator integrated with Trimaran.

Growth Plans

Adaptive Computing – While current processors are programmed through a fixed interface, adaptive computing is the design of computer hardware that incorporates dynamic configuration capabilities. This will facilitate the development of a wide variety of specialized systems by reusing a relatively small set of hardware designs.

- Weng Fai Wong, *National University of Singapore*
- Jean Vuillemen, *École Normale Supérieure*
- Wayne Luk, *Imperial College*
- Shay Ping Seng, *Imperial College*

Power Aware Computing and Compilation – Recent years have witnessed a power versus performance trade-off applied to large computing systems. The trend continues and is a major concern for processors and peripherals in portable equipment. Power aware computing and compilation is the innovation of new techniques for minimizing the power requirements of computer systems.

- Weng Fai Wong, *National University of Singapore*
- Peter Marwedel, *Universität Dortmund*

Microarchitecture Modeling Based on Microelectronics Foundations – With the evolution of System-On-a-Chip architectures and Adaptive Hardware, accurate modeling of the physical and practical limits that govern future opportunities in technology become more important. This effort will guide the design of systems in terms of die-size, timing and power.

- Jim Meindel, *Georgia Institute of Technology*
- Steve Nugent, *Georgia Institute of Technology*

Embedded Systems

All of the above areas involve continuing synergy with Hewlett Packard Laboratory and the Compiler and Architecture Research group.