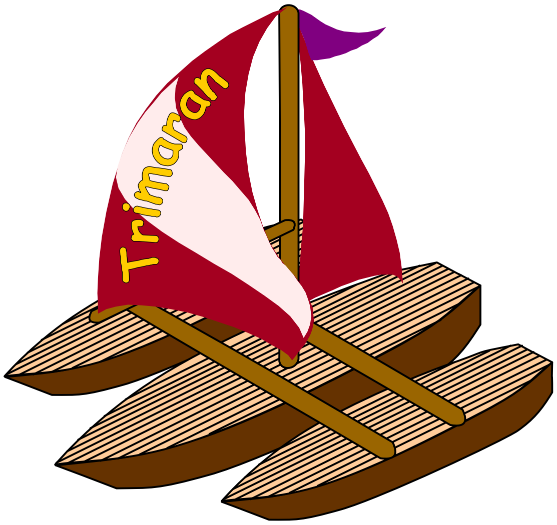


Trimaran

An Infrastructure for Compiler Research in Instruction Level Parallelism

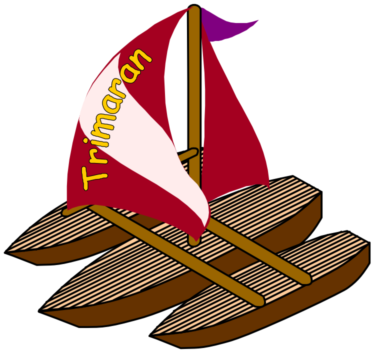
USER MANUAL

Hewlett Packard Laboratories, Compiler and Architecture Group

New York University, ReaCT-ILP Group

University of Illinois, IMPACT Group

Compiled by
Amitabh Nene, Suren Talla, Benjamin Goldberg, Hansoo Kim and Rodric M. Rabbah
New York University



Introduction

Welcome to the Trimaran Compiler Research Infrastructure. The Trimaran system is the result of many person-years of research and development in the Compiler and Architecture Research (CAR) group at Hewlett-Packard Laboratories, the IMPACT Group at the University of Illinois, and the ReaCT-ILP group at New York University. The system is distributed without charge for non-commercial use.

Trimaran is a compiler infrastructure for supporting state of the art research in compiling for Instruction Level Parallel (ILP) architectures. The system is currently oriented towards EPIC (Explicitly Parallel Instruction Computing) architectures, and supports compiler research in what is typically considered to be "back end" techniques such as instruction scheduling, register allocation, and machine-dependent optimizations.

This user manual is not an exhaustive listing of every feature of the Trimaran System. Rather, it is intended to get the Trimaran user started using the system as soon as possible. The manual tells you how to install and run Trimaran, gives a concise description of each component of the system, and contains pointers to the important files that should be perused in order to carry out compiler research using the system. The manual is organized as a collection of documents as follows:

1. This introduction
2. Installing and Running the Trimaran System
3. The Machine Description (mdes) facility.
4. The program Intermediate Representation (IR).
5. The simulation and performance modeling environment.

This manual is intended to be used in conjunction with the on-line documentation available in the Trimaran Graphical User Interface (GUI) and the Trimaran tutorial slides. As new material becomes available, it will be placed on the Trimaran web page, **www.trimaran.org**. Questions should be sent to **support@trimaran.org** and bug reports should be sent to **bugreport@trimaran.org.**

The Trimaran compiler infrastructure is comprised of the following components:

- A machine description facility, mdes, for describing ILP architectures.
- A parameterized ILP Architecture called HPL-PD.
- A compiler front-end, called IMPACT, for C. It performs parsing, type checking, and a large suite of high-level (i.e. machine independent) optimizations.

- A compiler back-end, called Elcor, 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) which 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. It is easy to use in its internal representation and its textual representation.
- 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.

The infrastructure is used for designing, implementing, and testing new compilation modules to be incorporated into the Elcor back end. These new modules may augment or replace existing Elcor modules, and may be the result of research in instruction scheduling, register allocation, program analysis, profile-driven compilation, etc.

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 (HPL-PD) space allows the user to experiment with machines that vary considerably in the number and kinds of functional units and register files and can vary in their instruction latencies. These machine configurations can be described both using the powerful machine description facility, mdes, 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 (Elcor) 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.

At this point, we recommend that you install and run the system, as described in the next section of the manual. As your use of the system becomes more extensive, you can then refer back to this manual.