

# RAMP: RESEARCH ACCELERATOR FOR MULTIPLE PROCESSORS

## Abstract for Hotchips 2006

Krste Asanović (MIT), Derek Chiou (UT Austin), James C. Hoe (CMU),  
Christoforos Kozyrakis (Stanford), Shih-Lien Lu (Intel), Mark Oskin (U Washington),  
David Patterson (UC Berkeley), Jan Rabaey (UC Berkeley), and John Wawrzynek (UC Berkeley)

### CONTACT INFORMATION:

Professor David Patterson  
635 Soda Hall  
Department of Electrical Engineering and Computer Sciences  
University of California, Berkeley  
Berkeley, CA 94720  
510-642-6587  
pattsrn@cs.berkeley.edu

Professor John Wawrzynek  
631 Soda Hall  
Department of Electrical Engineering and Computer Sciences  
University of California, Berkeley  
Berkeley, CA 94720  
510-643-9434  
johnw@eecs.berkeley.edu

# 1 Introduction

Processor architectures have crossed a critical threshold. Manufacturers have given up attempting to extract ever more performance from a single core and instead have turned to multi-core designs. While straightforward approaches to the architecture of multi-core processors are sufficient for small designs (2–4 cores), little is really known how to build, program, or manage systems of 64 to 1024 processors. Unfortunately, the computer architecture community lacks the basic infrastructure tools required to carry out this research. While simulation has been adequate for single-processor research, significant use of simplified modeling and statistical sampling is required to work in the 2–16 processing core space. Invention is required for architecture research at the level of 64–1024 cores.

Fortunately, Moore’s law has not only enabled these dense multi-core chips, it has also enabled extremely dense FPGAs. Today, one to two dozen cores can be programmed into a single FPGA. With multiple FPGAs on a board and multiple boards in a system, large complex architectures can be explored. To make this happen, however, requires a significant amount of infrastructure in hardware, software, and what we call “gateway”, the register-transfer level models that fill the FPGAs. We are building such a large-scale, FPGA-based emulation platform. We are also working to create a research community around our common hardware and software infrastructure, which will allow us to pool our resources to build a far more productive environment than we could have achieved individually.

We expect RAMP to provide multiple orders of magnitude speedup over software-based simulation. This speedup will provide a qualitative leap in the quality and range of future computer architecture research. Furthermore, since processors cores can run at 100 to 200 MHz, a large scale multiprocessor would be fast enough to run operating systems and large programs at speeds sufficient to support software research. We believe such a system would accelerate research across all the fields that touch multiple processors: operating systems, compilers, debuggers, programming languages, scientific libraries, and so on. Thus the acronym RAMP, for Research Accelerator for Multiple Processors.

Because RAMP is a research platform, we can offer capabilities not found in commercial multiprocessors. For example, RAMP could have reproducible behavior, where every processor’s memory references and interrupts would happen at exactly the same clock cycle on every run. RAMP would be valuable for many forms of software experiments, as well as aiding in software debugging.

We imagine that this infrastructure could be valuable to many projects, but here are a few representative examples:

- Testing the robustness of multiprocessor HW/SW under fault insertion workloads
- Developing thread scheduling and data allocation/migration techniques for large scale multiprocessors
- Developing and evaluating instruction set architectures for large scale multiprocessors
- Creating an environment to emulate a geographically distributed computer, with realistic delays, packet loss, and so on
- Evaluating the impact of 128-bit floating point on convergence of parallel programs
- Developing and field testing of hardware and software schemes for improved security.
- Recording traces of complex programs running on a large scale multiprocessor
- Evaluating the design of multiprocessor switches (serial point-to-point, distributed torus, fat trees)
- Developing data flow architectures for conventional programming languages
- Developing parallel file systems
- Testing dedicated enhancements to standard processors (a la Tensilica)
- Compiling DSP software directly into FPGAs

## 2 Technical Details

The RAMP vision is a scalable, multiboard FPGA-based system that would allow construction of up to a 1024-CPU multiprocessor. This size is an interesting target because many problems that are invisible at 32 processors and awkward at 128 become glaring at 1024. This scale challenge is true across the architecture, network, operating system, and applications disciplines. This shared artifact would consist of hardware and a collection of “gateway” (RTL) and software that members of the community would help create. We are seeding the cooperative effort using soft processor cores contributed from our industrial collaborators. For example, we are already working with cores from IBM, Sun, and Xilinx.

The goal is that our initial distribution of gateway and software would contain a complete configuration of a scalable multiprocessor populated with standard processor cores, switches, operating systems, and compilers. A user should be able to load binaries for the default cores and run them without modification. Hence, the default cores would implement a commercially available ISA that already has an open source implementation available. Our goal would be that once the base system is assembled and software installed, users could easily run complex system benchmarks and then modify this working system as desired, or start from the ground up using the basic components to build a new system. Users would release back to the community any enhancements and new hardware/software modules.

Rather than beginning the RAMP project by designing yet-another-FPGA-board, we are using the BEE2 design from the Berkeley Wireless Research Center (BWRC). The compute module consists of five Xilinx Virtex 2 Pro 70 FPGA chips each directly connected to four DDR2 240-pin DRAM DIMMs, with a maximum capacity of 4GB per FPGA. The four DIMMs are organized into four independent DRAM channels, each running at 200MHz (400DDR) with a 72-bit data interface. The BEE2 design is scalable to multiple compute modules through the use of 20 10Gb/s communication links per module.

A configured RAMP system models a collection of CPUs connected to form a cache-coherent multiprocessor. The emulated machine is called the *target*, and underlying FPGA hardware (e.g. BEE2) is the *host*. We have developed a design framework for RAMP, called RDL (RAMP Design Language), that allows flexible, modular, cycle-accurate emulation of target machines. The framework supports both cycle-accurate emulation of detailed parameterized machine models and rapid functional-only emulations. The framework hides changes in the underlying RAMP hardware from the module designer as much as possible, to allow groups with different hardware configurations to share designs and to allow RAMP modules to be reused in subsequent hardware revisions. The framework also does not dictate the hardware design language chosen by developers.

RDL is based on a few central concepts. A RAMP configuration is a collection of communicating *units*, where a unit is a relatively large number of gates. Example units include CPUs, coherence engines, DRAM controllers, I/O device interfaces, and network router stages. All communication between units is via messages sent over unidirectional point-to-point inter-unit *channels*, where each channel is buffered to allow units to execute decoupled from each other.

Based on the BEE2 host and RDL, we are simultaneously developing three RAMP target prototypes to help demonstrate the RAMP concept and work towards a public release. RAMP *red* is a multiprocessing system based on the hard PowerPC cores (part of the Virtex FPGAs) and a transactional memory system. We expect to demonstrate this system in early summer 2006. RAMP *blue* is a cluster of MicroBlazes (Xilinx soft CPU cores) connected through a message passing network. We plan to demonstrate this target by running scientific computing benchmarks sometime late summer 2006. RAMP *white* is a longer term project to develop a cache coherent shared memory multiprocessor.