

# Ant-32

## On the Design of a New CPU Architecture for Pedagogical Purposes

Daniel Ellard, David Holland, Nicholas Murphy, Margo Seltzer

Harvard University

WCAE '02

Funded by Microsoft and NSF

## Why Design a New Architecture?

- Too many architectures in our undergraduate curriculum!
  - Time-consuming to learn them all
  - Impossible to delve deeply into any of them
- Desire to use one platform in many courses
  - No suitable platform known to us
  - Other pedagogical machines target specific problems
- Our solution: a multi-purpose architecture

## An Earlier Experience - Ant-8

In 1996, we created Ant-8, a tiny 8-bit CPU

- Entire machine state fits on 24x80 screen!
- Now used in all our intro courses:
  - CS0 - Intro arch principles, asm prog
  - CS1 - Intro arch principles, asm prog, and write a simulator for Ant-8
  - Intro Arch - build in FPGA
- Too small/simple for advanced courses:  
compilers, OS

## A New Design - Ant-32

- Create a new 32-bit architecture
- Reuse successful concepts from Ant-8
  - Leverage student familiarity with Ant-8
- Add features needed for advanced classes
  - Extended instruction set
  - Ample address space and virtual memory
  - Exceptions, interrupts, external devices

## Guiding Principles

- ***“How are we going to explain this?”***
  - Documentation and related materials are the keys to a successful educational architecture
- Focus on the undergraduate curriculum
- Simplicity, Functionality, and Realism
  - 3-address RISC (very MIPS-like core)
  - TLB-based MMU
  - Austere exception architecture

## Instruction Set Decisions

- Bare-bones CPU functionality
  - Ant-8 plus relative branches
- “Convenience” instructions (like div)
- TLB instructions
- Exception-oriented instructions
- Instructions for compiler and OS writers
- Never-ending battle between what is nice and what is necessary!

# Virtual Memory System Decisions

- TLB-based model sufficient
  - Support for protection, virtualization
- 32-bit virtual, 30-bit physical addresses
- Four segments - two mapped, two unmapped
  - Unmapped segments provide simple flat address space - lets students write programs without any TLB setup at all (albeit in supervisor mode)

# Exception Architecture Decisions

- Definitely the most contentious area!
- Goal - simple exception handlers & glue code
  - Machine state easy to define
  - 8 dedicated registers used to preserve machine state during exception-handling
  - User traps and device interrupts are types of exceptions, handled with the same mechanism



## Features Omitted

- Features left unspecified (add if needed):
  - No hardware floating point
  - No explicit pipeline, multiple issue, etc
  - No explicit cache (although placeholders, like “uncacheable” bits in the TLB, do exist)
- Features designed out (very hard to change):
  - No hardware page-table (just a TLB)
  - Fixed page size

## Ant-32 Project Status

- Architecture spec, programming tutorial, assembler, debugger, and simulator available
- Additional teaching materials under development – lesson plans, assignments, lecture materials, texts ...
- Used in compiler class (at Boston College)
  - Not far into code generation yet
- Planned use in OS course next Spring
- Looking for partners/adapters/critics!

## Conclusions

- Ant-32 is designed to be useful in a variety of educational contexts.
  - Good illustration of basic concepts without unnecessary detail
  - Leverage knowledge between courses
- Ant-32 might completely replace Ant-8.
- We are studying the impact of using Ant-32 and will learn from our experiences.