On the Design of a New CPU Architecture

Ant-32

Funded by Microsoft and NSF

WCAE '02
Harvard University
Daniel Ellard, David Holland, Nicholas Murphy, Margo Seltzer

for Pedagogical Purposes
Why Design a New Architecture?

Our solution: a multi-purpose architecture

- Too many architectures in our undergraduate curriculum! - Time-consuming to learn them all - Impossible to delve deeply into any of them - Time-consuming to learn them all

Other pedagogical machines target specific problems - No suitable platform known to us - Desire to use one platform in many courses

Too many architectures in our undergraduate curriculum!

Why Design a New Architecture?
An Earlier Experience - Ant-8

In 1996, we created Ant-8, a tiny 8-bit CPU:
- Entire machine state fits on 24x80 screen!
- Entire Arch principles, asm prog, and
- Intro Arch - build in FPGA
- Intro Arch principles, asm prog, and
- CSCI - Intro Arch principles, asm prog
- CS0 - Intro Arch principles, asm prog
- CSCI - Intro Arch principles, asm prog
- CS0 - Intro Arch principles, asm prog
- CS1 - Intro Arch principles, asm prog
- Intro Arch - build in FPGA
- Too small/simple for advanced courses:

compilers, OS

Now used in all our Intro courses:
- Entire machine state fits on 24x80 screen!
- Entire Arch principles, asm prog
- Entire Arch principles, asm prog
- Entire Arch principles, asm prog
- Entire Arch principles, asm prog
- Entire Arch principles, asm prog
- Entire Arch principles, asm prog
- Entire Arch principles, asm prog

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

compilers, OS

Too small/simple for advanced courses:
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

WCAE'02 - http://www.ant.harvard.edu
Guiding Principles

• "How are we going to explain this?"

– Austere exception architecture
– TLB-based MMU
– 3-address RISC (very MIPS-like core)
– Simplicity, Functionality, and Realism
– Focus on the undergraduate curriculum
– Architecture
– The keys to a successful educational paradigm and related materials are

Guiding Principles
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!

What is necessary?

- Bare-bones CPU functionality
- Ant-8 plus relative branches
- "Convenience" instructions (like div)
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

(albeit in supervisor mode)

Unmapped segments provide simple flat address space - lets students write programs without any TLB setup at all!

Virtual Memory System Decisions
Exception Architecture Decisions

- Definitely the most contentious area!
- Goal - simple exception handlers & glue code
  - Machine state easy to define
  - Machine state during exception-handling
  - User traps and device interrupts are types
    - 8 dedicated registers used to preserve
  - Simple exception handlers & glue code

Mechanism

of exceptions, handled with the same
• Features designed out (very hard to change):
  - Fixed page size
  - No hardware page-table (just a TLB)
  - No hardware page-table (just a TLB)
  - No hardware page-table (just a TLB)
  - No hardware page-table (just a TLB)

• Features left unspecified (add if needed):
  - No explicit cache (although placeholders)
  - No explicit pipeline, multiple issue, etc
  - No hardware floating point
  - Features Omitted
Ant-32 Project Status

• Architecture spec, programming tutorial, assembler, debugger, and simulator available
• Additional teaching materials available
• Used in compiler class (at Boston College)
• Planned use in OS course next Spring – Not far into code generation yet

Looking for partners/adopters/critics!
Conclusions

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

and will learn from our experiences.