TRIPS CPU

James Halliday

October 15, 2009

What is it?

- Tera-op, Reliable, Intelligently adaptive Processing System
  - University of Texas, Austin
  - IBM
  - Intel
  - Sun

What is it?

- Goals
  - One TFLOP on a single processor by 2012.
  - Scale across hundreds or thousands of cores

How does it work?

- Explicit Data Graph Execution Instruction Set Architecture
- Hyperblocks
- - only 128 instructions
  - only 32 memory accesses
  - only 64 register accesses (32 read, 32 write)
  - one branch op (end)
- Compilers already know enough to build these structures

How does it work? (terms)

- Dynamic Issue *
  - Order of instruction execution is not known ahead of time
- Static Issue
  - Order of instruction execution is known ahead of time
  - It's really hard to know whether or not memory is in cache
  - Waiting for memory to appear in cache holds up everything else
- Dynamic Placement
  - Dependency analysis performed at runtime
- Static Placement *
  - Dependency analysis performed at compilation
  - Programs are only compiled once, but run many times

How does it work?

- Other Approaches to Parallelism
  - Superscalar
    - Dynamic Placement
    - Dynamic Issue
  - Itanium
    - Very Long Instruction Words
    - Static Placement
    - Static Issue
  - SIMD
    - Hard-Coded Vector Sizes
    - Extra Circuit Complexity

How does it work?

- Static Placement
- Dynamic Issue
- TRIPS is lazy!
- Local dependencies are optimized by the compiler, not the CPU

Why should I care?

- Exploit massive parallelism in sequential programs!
- Hundreds or thousands of identical cores on the same chip
- Compilers are better at figuring out mundane details than you are

Why should I care?

- Speaker: James Halliday
- Subject: TRIPS CPU
- What it is
  - experimental, lots of identical cores
- How it works:
  - hyperblocks, static placement, dynamic issue
- Why you care:
  - compiler figures out parallelism details on thousands of cores