MultiScalar
Your questions

• How does register allocation work?
• How bad is latency to and from the ARB?
• How scalable is the ARB?
The quest for parallelism

- Single threads have a little bit of ILP
- We want MORE!
- Multithreaded programming is hard
  - Locks are tricky
  - Often, statically available parallelism is just scarce
  - Dynamic data dependences are unpredictable
MultiScalar idea

- Use SW + HW to divide the program into pieces
  - What should the HW look like?
  - How should the SW express the pieces?
- Speculatively run consecutive pieces in parallel
  - Which pieces?
- Clean up the mess
  - This is the tough part.
MultiScalar tasks

- Any dynamic sequence of instructions can be a task
- Arbitrarily large!
- Arbitrarily small!
- Any of number of exits!
- Function calls!
MultiScalar
MultiScalar key points

- Coarse-grain control
- Program decomposition into tasks
- Value forwarding
- Memory disambiguation
- Very, very large instruction window
Decomposing programs

• Chop up the program.
• In principle, you can do this anywhere
  • In practice it’s harder
  • Functions?
  • Entire loops?
• What to large speculative chunks mean?
Tasks

for (indx = 0; indx < BUFSIZE; indx++) {
    /* get the symbol for which to search */
    symbol = SYMVAL(buffer[indx]);

    /* do a linear search for the symbol in the list */
    for (list = listhd; list; list = LNEXT(list)) {
        /* if symbol already present, process entry */
        if (symbol == LELE(list)) {
            process(list);
            break;
        }
    }

    /* if symbol not found in the list, add to the tail */
    if (!list) {
        addlist(symbol);
    }
}

Figure 3: An Example Code Segment.

Figure 4: An Example of a Multiscalar Program.
Coarse grain control

• Aggressive next-task prediction
• Where have seen this before?
• They claim it’s easier than branch prediction -- hmm...
Value forwarding

- Values need to get to speculative threads fast
- Problems:
  - What are the outputs? -- mask
  - What are the inputs? -- mask
  - Which version to use? -- extra bits on the instruction.
Memory disambiguation

• Memory is the hard problem.
• They use an address resolution buffer (ARB)
• Any problems with this?
Figure 1: A Possible Microarchitecture of a Multiscalar Processor.
Large instruction window

Point of Search

ACDE

Window

ACFH

Point of Search

Window
In Context

- MultiScalar is very influential and extremely ambitious.
- Spawned much work in speculative threading on more reasonable architectures.
- If we evaluate it in terms of modern technology, how does it hold up?