The Lazy Student and the Computer Architect

ISPASS-05 Technical Panel

March 21, 2005

Doug Burger
Computer Architecture and Technology Laboratory
Department of Computer Sciences
The University of Texas at Austin

Discussion of Trends

- Frequency is obviously slowing
  - 40% per year since 1990, half due to pipelining, half devices
  - Effects: froze ISAs, particularly in academic research
  - Exacerbated memory latencies and dynamic power limits more quickly
  - Pipelining pretty much done, devices may slow due to leakage
  - Clocks eventually going the other way?
- Power is now a constraint just like area
  - Here is your power and area budget, how much performance can you get?
- Leakage will create new problems and opportunities for architects
  - Transistors are no longer free each generation
  - May have 10B on a chip, but only 500M can be powered up
- Wire delays continue to get worse
- Don’t believe uniprocessor performance claims
  - Who saw the web coming?

The times, they are a’changing

- Big changes underway in our field, for all the usual reasons
  - Wires, power, complexity, workloads, costs, reliability, memory latencies, blah, blah, etc., etc.

- How should we evaluate future directions given all of these strains?

Follow the Lazy Student!

- What do students do?
  - Learn from the past ... so should we when projecting new models.

- What do lazy people do?
  - Minimize effort ... so should we when designing new interfaces.

Learn from the Past

- We have to ask ourselves when re-proposing an old model ... what has changed?
  - How often do smart people work on a hard problem for two decades, fail to solve the problems, and then all of a sudden the problem is solved in the conventional context?
  - Almost never!
- Good examples from recent history:
  - VLIW - what changed to make IA-64 viable whereas VLIW didn’t work the first few times?
  - CMPs - parallel programming has always been just around the corner
    - Every time the market has voted, it has voted for uniprocessors!
    - 90M-transistor, single-core Pentium IV, or 450 8086 processors?
    - What is different now?

It’s Good to Be Lazy

- Key goal is to minimize effort
  - Programmer laziness is always underestimated (not pejorative)
  - Architect’s proposing programming models is dangerous
  - Compiler effort often underestimated by microarchitects
  - Hardware effort now driven by power
  - DId we really hit a complexity wall in design effort (probably close)
- Future systems must renegotiate interfaces to minimize effort at multiple levels
  - VLIW was too much compiler effort (wrong burdens on the compiler)
  - Wide-issue RISC/CISC too much hardware effort (power)
  - Leakage will create new metrics for effort (things just being on)
  - CMPs are too much effort for most programmers
  - We must re-negotiate these interfaces!
  - Concurrency is key, but not the types that people typically think! (i.e. what is single thread performance?)

Conclusions

- We have to get concurrency as easily as possible
- Must re-negotiate compiler/hardware interfaces significantly
  - Anything done in the compiler is energy-free at runtime!
- Must re-negotiate programming models/parallel architecture interfaces
  - Lots of interesting work
  - Speculative locking, transactions
  - Vectors/threads/speculation/SIMD/Other and mix!
- General-purpose computing will continue
  - The markets under it may shift
  - People will find ways to use the performance
- Need to convey the need for innovative research to the wider community