Issues in Computer Architecture and Microarchitecture …for future computing machines

Yale N. Patt
The University of Texas at Austin

Version 2009

Introduction and Focus

* *Overriding consideration:*

  What does it cost?
  What is the benefit?

* *Global View*

  --Global vs. Local transformations

* *Microarchitecture view*

  --The three ingredients to performance

* *Physical view*

  --Partitioning
  --Power consumption
  --Wire delay

Problem

Algorithm

Program

Instruction Set Architecture (ISA)

Microarchitecture

Circuits

Electrons

The Triangle

* *Only the algorithm knows the programmer*

  -- pragmas
  -- pointer chasing
  -- partition code, data

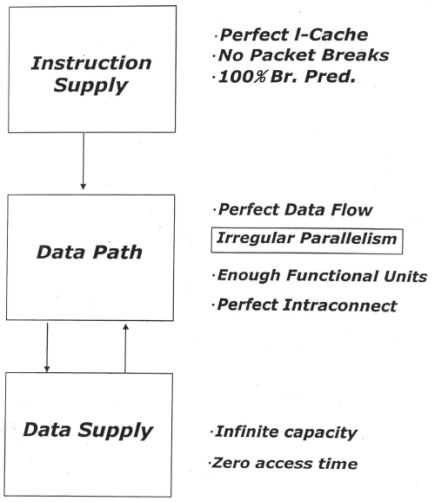
* *Only the compiler knows the future (?? :-)*

  -- Predication
  -- Prefetch/Poststore
  -- Block-structured ISA

* *Only the hardware knows the past*

  -- Branch directions
  -- Cache misses
  -- Functional unit latency

Microarchitecture (The Requirement)



Speculation

- ***Why good? – improves performance***
- ***How? – Guess***
  - ***Branch prediction***
  - ***Way prediction***
  - ***Data prefetching***
  - ***Value prediction***
  - ***Address prediction***
- ***Why bad? – consumes energy***

The memory problem

- ***Simply: off-chip, on-chip access time imbalance (assuming bandwidth is not a problem)***
- ***What to do about it***
  - ***MLP replacement policy***
  - ***DIP replacement***
  - ***VWay cache***
  - ***Runahead execution***
  - ***Address, value prediction***
  - ***Two-level register file***
  - ***Denser encoding of instructions, data***
  - ***Better organization of code, data***

Embedded Processors vs. General Purpose

- ***What is different from what we've said?***
- ***Easier to design wholistically***
  (limited purpose, special purpose)
- ***Greater use of ASICS***
  (The x + superscalar syndrome)
- ***Why VLIW***
- ***Partitioning***
- ***Memory latency***
- ***Reconfigurability ??***

Design Points

- ***-- Performance***
- ***-- Reliability***
- ***-- Availability***
- ***-- Cost***
- ***-- Power***
- ***-- Time to Market***

Design Principles

- ***Critical path design***
- ***Bread and Butter design***
- ***Balanced design***

Role of the Architect

-- Look Backward (Examine old code)

-- Look forward (Listen to the dreamers)

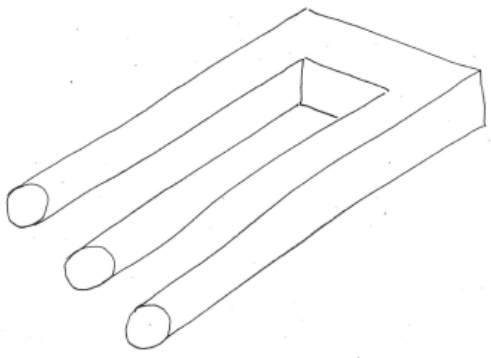
-- Look Up (Nature of the problems)

-- Look Down (Predict the future of technology)

Some "new" technology issues

- ***Wire delay***
- ***Power, energy***
- ***Soft errors***

Finally, people are always telling you: Think outside the box



I prefer: Expand the box

Agents of Evolution

-- *Performance*

-- *Bottlenecks*

-- *Good Fortune*

Evolution of the Process

- *Pipelining*
- *Pipelining (with hiccups)*
- *Wide-issue*
- *Serious Branch Prediction*
- *Speculation ...ergo: Recovery*
- *Trace Cache*
- *SMT, SSMT*
- *Soft Errors*
- *L2 misses*
  (e.g., run-ahead, KiloInstruction Processors)

A Few Specifics

- *HPS – expanded on Tomasulo*
- *SMT – expanded on Burton*
- *Perceptron predictor – expanded on Widrow/Rosenblatt/etc.*