



Lots and Lots of Threads

An Argument for Exploiting the
Oncoming Explosion of Available
Thread-level Parallelism

Chuck Moore
October 02, 2000



IBM

Thread-level Parallelism

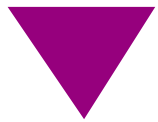
Three Key Observations:

The amount of Instruction-level parallelism is, in practice, relatively limited.

High Frequency operation has contributed greatly to recent performance gains, but the trends are likely to slow down within the next decade.

The amount of available Thread-level Parallelism will increase dramatically over the next decade.

**Companies that are best poised to exploit Thread-level Parallelism
will win in the market**



Limited Amount of Instruction-level Parallelism

Option One: Wide Issue Machines using Existing RISC Architectures

Diminishing returns

- ▶ 8 issue machine likely to be >2X size of 4 issue machine (for 20-30% IPC)
- ▶ Especially true on Server workloads

Likely to compromise operating frequency

- ▶ Large design will suffer more from increasing wire delays
- ▶ More circuits to optimize probably means less focus on a per circuit basis

Complexity will require more people and more time

- ▶ Verification completeness highly questionable

IPC Benefits are typically negated by cycle time and time-to-market losses

Option Two: New (Underlying) Core Architecture

New incompatible architecture - bad idea from the start (transitioning burden)

- ▶ Perhaps consider if >>2X IPC performance possible - but, invention required!

Mimic new architecture "under the covers"

- ▶ Likely to introduce significant "creeping" complexity to yield performance gain
- ▶ Still requires long-term transition plan to achieve full potential

On balance, a "knee of the curve" superscalar design is tough to beat



Frequency Growth will Slow Down

Two Primary Degrees of Freedom:

1. Pipeline / Machine Organization

Early 90s: 25-30 FO4 designs, moderate superscalar, "structured custom" design

- ▶ Achieved 300-500 MHz in 0.25u technology
- ▶ Frequency compromised by drive for high IPC

Late 90s: 18-22 FO4 designs, "knee of the curve" superscalar, full custom design

- ▶ Achieved 1000-1200 MHz in 0.18u technology
- ▶ Power consumption and wire delay became significant considerations

Early 00s: 10-14 FO4 designs possible, full custom design, aggressive circuits

- ▶ Wire delays force extraordinary amount of planned partitioning
- ▶ Degree of customization will probably inhibit "wide machines"

2. Process Technology

**Reasonable optimism to continue trends down to 0.07u . . . very tough beyond that
Revolutionary technology breakthroughs possible?**

- ▶ Maybe, but development expense is huge
- ▶ Also, early/mid-life reliability factors become primary consideration

Asynchronous design techniques?

- ▶ Maybe, but benefit is limited. Also, verification would be a major challenge.

Quantum / Molecular computing?

- ▶ Maybe, but machines likely to be very simple

A 10 FO4, full custom CMOS design in 0.07u can land in the 5 GHz range



Expanding Amount of Thread-level Parallelism

This has traditionally been a "chicken and the egg" problem

- ▶ But . . . access to SMP machines is no longer an issue
- ▶ Universities, start-ups and individuals have access and see the opportunity

OS & Application structuring for NUMA/Clustering is a step in this direction

- ▶ Partitioning, locality and attention to communication/synchronization issues
- ▶ Partitioned database technology is already available

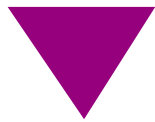
Consider, for a moment, the internet as a massive multicomputer

- ▶ Lots of research and practical applications beginning to appear
- ▶ Beowolf clusters, SETI@home, largest prime number
- ▶ Algorithms being re-thought to exploit large scale threading

The Performance Leverage of TLP is Huge

- ▶ Algorithmic performance leverage (not just incremental)

Prediction: Once this cycle truly engages, available thread-level parallelism will increase exponentially



Possible Implications of TLP

Large, complex designs may be replaced by small, flexible designs

- ▶ Use multithreading to help manage latency
- ▶ Integrate multiple cores on a single die
- ▶ Include mode for "replicated pair checking" for excellent error handling
- ▶ Trickle-down into high-end SOC and embedded applications

System Architecture enhancements designed to exploit TLP

- ▶ On-chip "lock box" to speed local sharing and synchronization
- ▶ Expand MMU to include "Network Management" functions
- ▶ System action queue and "system ISA"
 - ▶ Page functions (copy, move, zero, compare, etc)
 - ▶ Object functions (copy, move, prefetch, etc)
 - ▶ Programmable prefetch engines
 - ▶ Dynamic encryption and/or compression
 - ▶ more . . .

**The Applications Development Platform of the Future Should
Focus on Enabling Thread-level Parallelism**