Threads,
Threads,
Threads

David Golombek
Architectures Anonymous

What is MTA?

* Tera's Multi-Threaded Architecture

* Emphasis on very high numbers of threads,
  and low switching time.

* True shared memory model

* Simplified superscalar+VLIW instruction capabilities

* Removes all data cache, 8K L1+ 2M L2 instruction caches

What is SMT?

* Compaq's Simultaneous Multi-Threading

* Will show up in the 21464 (~2003)

* 8-wide superscalar, with 4 threads

* 1 primary thread, others are run when possible

* Keeps all caches, adds L1 cache/thread

Advantages and Disadvantages of SMT and MTA

* MTA simplifies chip by removing caches
  but therefore requires more threads

* MTA requires entire new design, SMT is incremental

* MTA is much more scalable

* MTA has simpler resource contention handling

How many threads are needed on MTA?

* Average of 70 cycle latency on memory load

* Without lookahead, 70 threads needed

* 8 concurrent memory references/thread

* With lookahead, ~20 threads needed

* Each chip supports 128 threads

How to get that many threads

* Normal C/Fortran code with slight modifications

* Compiler extracts some parallelism automatically

* Programmer points out other parallelism manually

* When in doubt, run more programs

Lightweight thread creation

* Thread creation is in user-space with chip support

* No context switching

* Status Reg, 32 GPRs, 8 SPRs (Target Regs)

* Only takes two instructions for minimal thread,
  20-30 for full

Convincing programmers to use threads

* Give them good tools

* Traceview: graphical processor utilization tool

* canal: compiler analysis tool

* tdb: Tera’s version of GDB

Convincing your compiler to find threads

* Compiler directives and automatic optimizations

* Good at loop nested optimization/parallelization

* Parallelization of reductions and recurrences

* Canal helps you figure out what it did

* Annotation system to "teach" compiler

Hardware resource usage with MTA

* Removing caches simplifies chip design greatly

* Thread selection done in hardware, but is very simple

* Add 4 bits per memory address to support synchronization

* Lots of registers

Hardware resource usage with SMT

* Added 10% to chip size, to get 125-175% speedup

* Added L1 per thread

* Expected high resource contention

Where does *T come into this?

* Message passing dataflow architecture

* Separate Data and Synchronization processors

* Similar multi-threaded design

What is CMP, and how does it play in?

* Chip Multi-Processor

* Multiple separate cores on one chip

* Use existing threading methodology to split work between chips

* Easy to design in hardware, each core is very simple

* Low degree of resource sharing

Merging and combining these architectures

* In the end, compiler advances may decide which wins

* SMT and CMP fairly compatible

* EPIC is resource intensive

Why did Tera buy Cray?

* Combine sales forces and manufacturing processes

* Get the employees, technology (especially software tools)

* Get the installed base

* Relatively cheap price (~$100M ?)

Merging Tera and Cray

* Small company buying much bigger

* Hardware is likely to remain independent for a while

* Biggest immediate impact will be on software