Today's Plan

Success stories

The Instruction Set Architecture (ISA)

- Look at microprocessor trends over the years
- What can we learn from this success story?

Examples of Success in Computer Architecture

The Instruction Set Architecture (ISA)

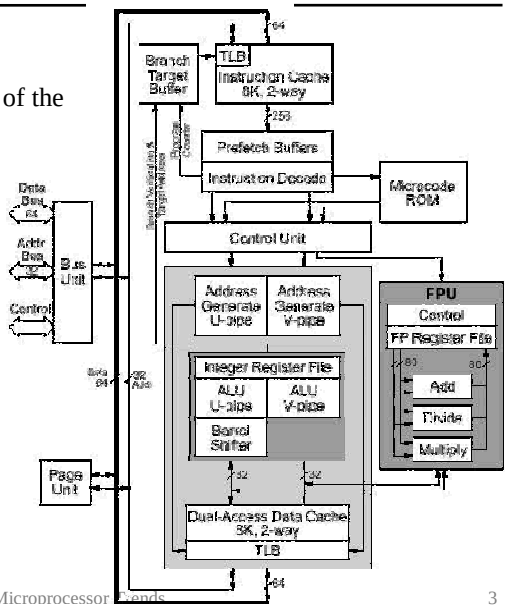
- The ISA is an example of a successful parallel abstraction
- Today we'll look at microprocessor trends over the years
- What can we learn from this success story?

Hidden vs. exposed technology

Parallelism in Computer Architecture

Hardware exploits parallelism

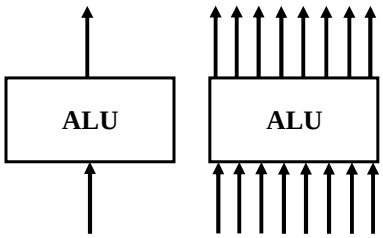
- Consider the block diagram of the Pentium processor



Parallelism in Computer Architecture

Hardware has long exploited parallelism

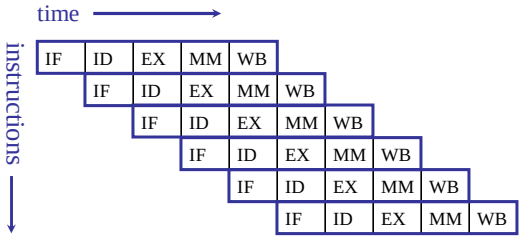
- Bit-serial ALUs
- Bit-parallel ALUs
- Pipelined microarchitectures



Pipelined Microarchitectures

Divide instructions into stages

Stage 1		Stage 2		Stage 3		Stage 4		Stage 5
Instruction Fetch	⇒	Instruction Decode & Register Fetch	⇒	Execute	⇒	Memory Access	⇒	Register Write-back
IF	⇒	ID/RF	⇒	EX	⇒	MEM	⇒	WB



A common form of task parallelism

- Increases latency of an individual instruction
- Increases throughput—ideally completes one instruction per cycle

Parallelism in Computer Architecture

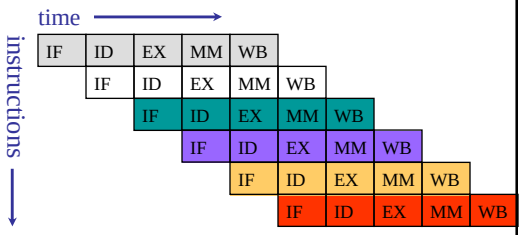
Hardware has long exploited parallelism

- Bit-serial ALU's
- Bit-parallel ALU's
- Pipelined microarchitectures
- Out of order execution

How does out of order execution help?

Program order

Issue order



Parallelism in Computer Architecture

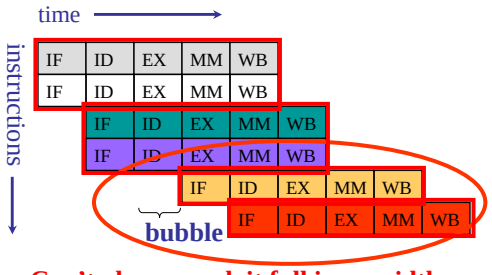
Hardware exploits parallelism

- Consider the block diagram of the Pentium processor

Hardware has long exploited parallelism

- Bit-serial ALU's
- Bit-parallel ALU's
- Pipelined microarchitectures
- Out of order execution
- Superscalar execution
- Multithreading

Issue multiple instructions per cycle

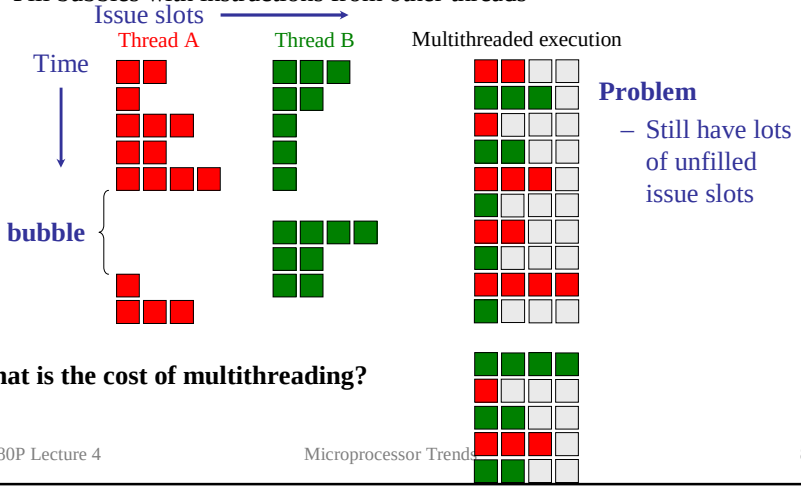


Can't always exploit full issue width

Multithreading

Key idea

- Bubbles in the pipeline represent performance loss
- Fill bubbles with instructions from other threads



Problem

- Still have lots of unfilled issue slots

What is the cost of multithreading?

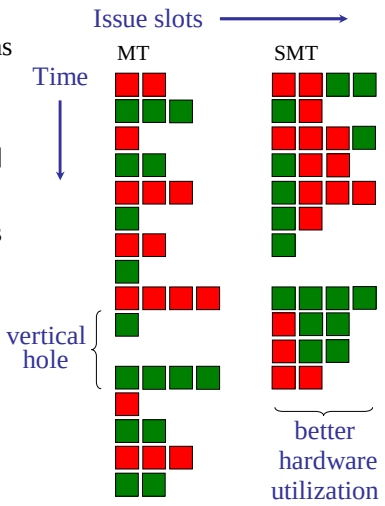
Improving Hardware Utilization

Limitation of Multithreading

- At each cycle, issues instructions from any one thread

SMT [Tullsen, Eggers, Levy ISCA95]

- Simultaneous Multithreading
- At each cycle, issue instructions from multiple threads
- The different threads share various resources, such as the cache
- Intel has adopted SMT and dubbed it hyperthreading



What is the cost of SMT?

Parallelism in Computer Architecture

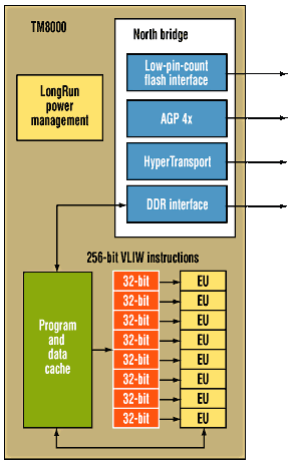
Hardware has long exploited parallelism

- Bit-serial ALU's
- Bit-parallel ALU's
- Pipelined microarchitectures
- Out of order execution
- Superscalar execution
- Multithreading
- SMT
- VLIW

VLIW

Very Long Instruction Word

- ISA explicitly encodes parallelism
- Instructions are wide so they can control multiple functional units
- Statically scheduled
  - Gives more control to the software
  - Popular for embedded processors because less power is expended on control



Transmeta TM8000 Core

Vector Processors– A Success Story

Goals

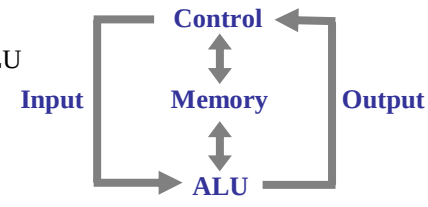
- Reduce instruction fetch bandwidth
- Produce streams of data that keep the processor busy

Vector processors

- One instruction controls all elements of a vector
  - eg. `VectorAdd`

The von Neumann Bottleneck?

- Bandwidth between memory and ALU
- Bandwidth between memory and Control



Vector Processors

Performance

- Cray: 1 or 2 orders of magnitude improvement for vector operations

Heyday

- In the mid-80's there were 20 manufacturers of vector processors

Several Years Ago: Endangered Species

- Only the Japanese made vector supercomputers
  - "Are we losing our expertise with vector processors?"

The Big Picture

Hardware has long exploited parallelism

- Bit-serial ALU's
- Bit-parallel ALU's
- Pipelined microarchitectures
- Out of order execution
- Superscalar execution
- SMT
- VLIW
- Vector support (MMX)
- Multi-core (aka CMPs)

Increasing granularity of parallelism

Which of these expose parallelism to through the ISA?

Conclusions

- For years the ISA was able to hide parallelism
- We're no longer able to hide the parallelism, so no more free lunch
- Is hidden complexity always good?

Hidden vs. Exposed Technology

How many computers do you own?

Some of the best uses of computers are hidden

- Cell phones
- Digital cameras
- Toasters
- Rear view mirrors
- Braking systems
- . . .

When should we expose technology? When should we hide it?