Computer Science 324
Computer Architecture
Mount Holyoke College
Fall 2007

Topic Notes: Modern Architecture Theme: Parallelism

The increases in processing power for decades have come from, at least in part, faster and faster clock speeds.

- we have seen in this course some of the reasons for limitations – gate delays
- smaller components means shorter gate delays, allowing shorter clock cycles and faster processors
- as we approach the physical limitations of the sequential processor, performance gains are coming more and more from the exploitation of parallelism
- there are many ways to expose native concurrency and introduce explicit parallelism to our processors, and we'll look at a few today

---

Intel Pentium Parallel Extensions

You may have heard of the MMX (and AMD's 3DNow!, and more recently SSE, SSE2, SSE3) extensions to the Intel Pentium core.

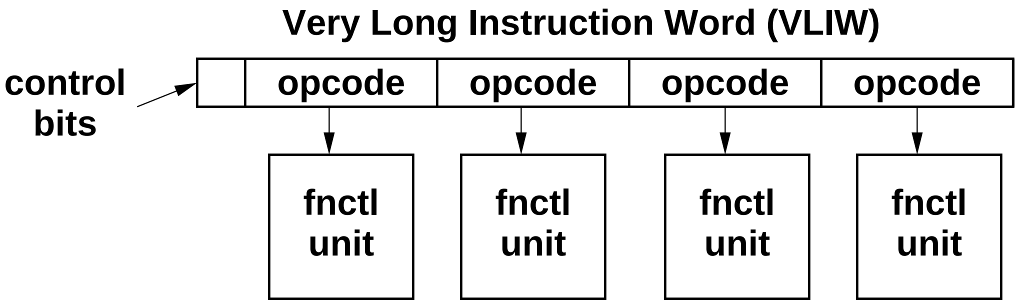
- SSE = Streaming SIMD Extensions, SIMD = single instruction multiple data
- Very simple idea to support arithmetic on short operands: cut the carry lines – makes it possible to manipulate 8 bytes (64 bits) independently but simultaneously in a single instruction
- packing and unpacking instructions are needed
- relatively few changes to the ALU, and we get the manipulation of two RGB+α pixels in a single operation

  ```
  rrrrrrrr gggggggg bbbbbbbb alphalph rrrrrrrr gggggggg bbbbbbbb alphalph
  ```

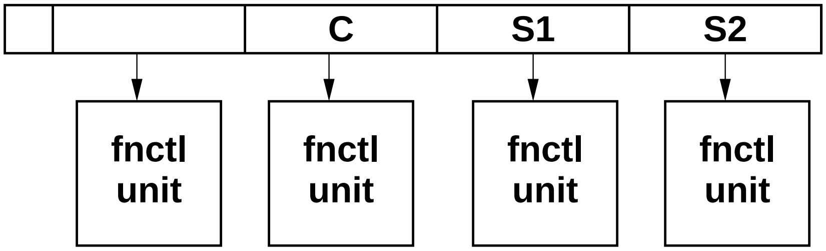
  treat as 8, 8-bit numbers
- several modes provided allow the ALU to consider its input as 8 independent bytes or 4 independent 2-byte values or 2 independent 4-byte values
- makes programming more challenging: consider conditional pixel modification

Explicitly Parallel Instruction Computers

The *explicitly parallel instruction computer* (*EPIC*) or *very long instruction word* (*VLIW*) machine: Yale's ELI and Intel's Itanium



- New architectural approach
- An instruction or *molecule* is made up of several concurrently executed *atoms*
- Each atom is assigned to a separate *functional unit*
- Processor has 256 registers, each atom gets its own copy of registers that are *committed* only when the atom is retired
- A few (8 of 128) bits of the molecule govern the execution of the atoms
- Speculative execution: avoid conditional branch overhead—execute `then` *and* `else`, but commit only one (Disadvantage: some work is guaranteed to be wasted)



```
if (C) {
  S1;
  ...
}
else {
  S2;
  ...
}
```

- Very complex programming – meant to be done by compilers, not people

Multicore Architectures

The recent approach involves replicating processing "cores" on the same chip that traditionally held a single processor.

This has changed the nature of the increases in processing capabilities:

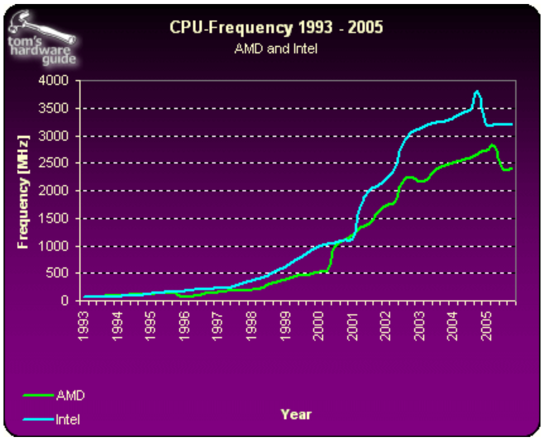


Figure used with permission from article *The Mother of All CPU Charts 2005/2006*, Bert Töpelt, Daniel Schuhmann, Frank Völkel, Tom's Hardware Guide, Nov. 2005, `http://www.tomshardware.com/2005/11/21/the_mother_of_all_cpu_charts_2005/`

We will look at the architecture – programming these is a nightmare for another day.

Intel/AMD Multicore

Intel and AMD have both introduced a series of chips that contain multiple processing cores.

The Intel Core Duo:

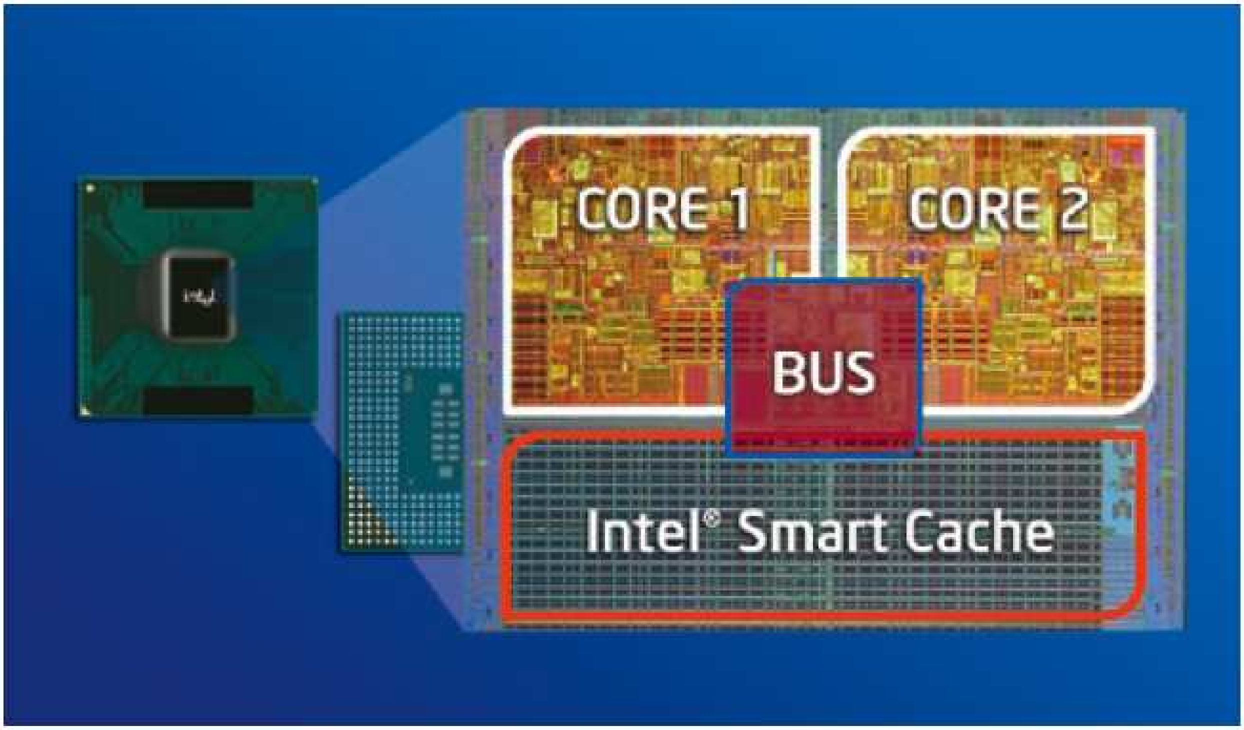


Image from Intel Core Duo Processor product brief

- Independent copies of ALU, registers, L1 cache

- Processors on the same chip share L2 cache
- Up to the operating system to schedule processes/threads to keep each core occupied

Current Intel and AMD processors are now 2- or 4-cores.

- How to deal with the memory latencies?
- Nice feature: more fine grained power management
- Intel has demonstrated an 80-core chip!

Cell Broadband Engine

IBM, Sony, and Toshiba collaborated on the recent *Cell* architecture.

- Consists of one or more PowerPC Processor Elements (PPEs) that are like traditional processors, and several Synergistic Processor Elements (SPEs) that are simpler processors that only perform work as assigned to them by PPEs
- Instructions and the data they manipulate are bound together in an *apulet*
- A *cell* is a hierarchically structured "bundle of control and streaming processor resources" or scalable *processing element*
- Apulets can be arbitrarily assigned to cells
- More intense computation is performed by adding more cells to the pool
- Currently used in Playstation 3

Graphics Processing Units

Computer graphics has driven the development of modern SIMD (single instruction multiple data) processors used as Graphics Processing Units.

- Graphics computations are often applied to a group of pixels at the same time – hence the SIMD approach – you can process many pixels at once (typically 128), but you have to do exactly the same operation on each
- Typically restricted to the single-precision floating-point operations needed for graphics
- Focus on maximizing "frames per second"
- Operations use graphics terminology: "pixel shaders" or "vertex shaders"

- But...deliver hundreds of gigaflops of performance where traditional CPUs are in the tens at best
- People have noticed this performance and have harnessed this computational power for non-graphics applications
- GPU producers have noticed this interest and are now providing better programming capabilities and double-precision operations (needed for most serious scientific calculations)