Slide 1

Administrivia

- Review sheet on Web soon. Extra-credit problems on Web soon. To be due no earlier than May 10. (So you have some time.)
- I will do a review session during the reading days. We'll pick a time next class or Friday.

Slide 2

Questions from Minute Essays

- "Does MIPS always use pipelining? do other languages?" (No. Probably mean "other architectures", and also "no", though probably most current implementations do.)
- "How does it decide what to cache?" (For direct-mapped cache, no choice possible/necessary. For fully associative cache, cache holds most-recently-used data.)

Slide 3

A Little More About Virtual Memory — Key Ideas

- Each process (application, user, etc.) has "address space" — range of possible addresses.
- This "virtual memory" is mapped onto a combination of real memory and disk ("swap space"). Goal is for real memory to act like a cache, holding data used recently/frequently.
- Virtual/real memory divided into "pages". Each process has a "page table" to keep track of its pages.

Slide 4

A Little More About Virtual Memory — Key Ideas, Continued

- When a program references a (virtual) address, hardware translates this to a real address using page table. To make this fast, page table entries cached (TLB).
- If translation is impossible, "page fault" exception. Operating system must then decide — error, or page on disk? and then take appropriate action.
- Deciding which pages go on disk, managing transfers between disk and memory, etc., etc. — "memory management" component of operating system.

Slide 5

Interfacing Processors and Peripherals

- Processor is connected both to memory and to I/O devices by "bus" (figure 8.1). Bus consists of "control lines" and "data lines". Part of role of control lines is to say what's on data lines (e.g., memory address or actual data to write). Simplified examples of use in figures 8.7 and 8.8.
- Can have processor-memory buses (connecting processor and memory only), I/O buses (connecting processor and I/O devices only), and backplane buses (connecting all three). Real systems can have a single backplane bus or multiple buses (figure 8.9).
- Buses can be synchronous (appropriate for processor-memory bus) or asynchronous (appropriate for I/O bus). Latter requires "handshake protocol" to coordinate things.
- Buses can have single "bus master" (processor) or allow multiple masters. Latter requires "arbitration" protocol.

Slide 6

Communicating with I/O Devices

- To tell I/O device what to do, processor must write "commands" to device controller. Two basic ways to do this — "memory-mapped I/O" (to processor, looks like regular memory access, but requests routed to I/O devices rather than memory) and special I/O instructions. Similar considerations for getting status/data from device.
- Since I/O is slow compared to processor speed, and time it takes may vary, must have some way to determine that an I/O operation is done. Two basic ways to do this — polling (keep asking device) and interrupts (have device generate exception/interrupt). Recall that interrupt causes transfer of control to fixed location, usually holding operating-system code.
- Transfer of data (between memory and device) can be under control of processor or done by separate DMA ("direct memory access") controller.

Slide 7

Communicating with I/O Devices, Continued

- In all but simplest systems, I/O is managed by operating system — allows sharing of physical resources among applications, relieves applications of low-level details.

Slide 8

Minute Essay

- None — sign in.
- Reminder: Homework 7 due by 5pm today.