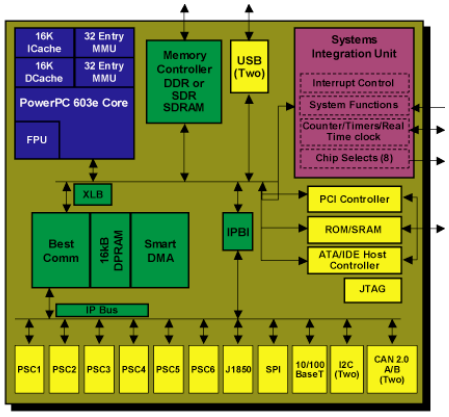
Lecture 22: Interconnects & I/O

- Administration
  - Take QUIZ 16 over P&H 6.6-10, 6.12-14 before 11:59pm
  - Project: Cache Simulator, Due April 29, 2010
  - NEW OFFICE HOUR TIME: Tuesday 1-2, McKinley
  - Exams in ACES 3.422 with Gem Naivar
- Last Time
  - Reliable and Available Storage
    - Memory technology, Disk, RAID, Flash
- Today
  - How are chips connected to Memory, Disk, & other devices?
  - How does the system perform I/O operations?

Example Networks on chip
Freescale: MPC5200 2003 Telematics chip

Telematics Domain

- Embedded processor for cars
- Combines wireless, graphics, audio, & GPS for integrated applications
- Embedded Linux OS



Interconnecting Components

- Need interconnections between
  - CPU, memory, I/O controllers
- Bus: shared communication channel
  - Parallel set of wires for data transfer and synchronization
  - Becomes a bottleneck
  - On chip does best with known latencies
- Performance limited by physical factors
  - Wire length, number of connections
- Modern alternative
  - high-speed serial connections with switches
  - On Chip Networks

Bus Types

- Processor-Memory buses
  - Short, high speed
  - Design is matched to memory organization
- I/O buses
  - Longer, allowing multiple connections
  - Specified by standards for interoperability
  - Connect to processor-memory bus through a bridge

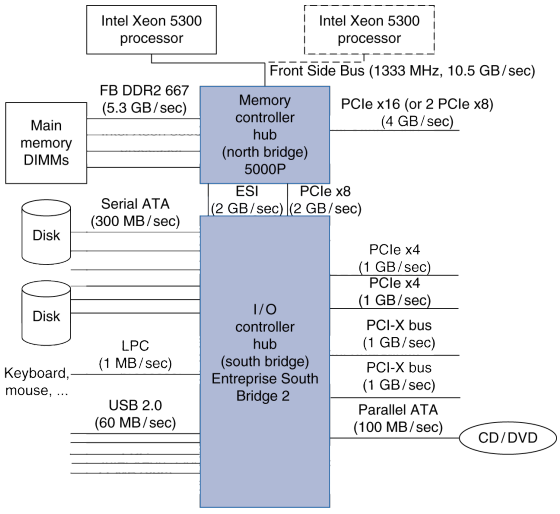
Bus Signals and Synchronization

- Data lines
  - Carry address and data
  - Multiplexed or separate
- Control lines
  - Indicate data type, synchronize transactions
- Synchronous
  - Uses a bus clock
- Asynchronous
  - Uses request/acknowledge control lines for handshaking

I/O Bus Examples

	Firewire	USB 2.0	PCI Express	Serial ATA	Serial Attached SCSI
Intended use	External	External	Internal	Internal	External
Devices per channel	63	127	1	1	4
Data width	4	2	2/lane	4	4
Peak bandwidth	50MB/s or 100MB/s	0.2MB/s, 1.5MB/s, or 60MB/s	250MB/s/lane 1×, 2×, 4×, 8×, 16×, 32×	300MB/s	300MB/s
Hot pluggable	Yes	Yes	Depends	Yes	Yes
Max length	4.5m	5m	0.5m	1m	8m
Standard	IEEE 1394	USB Implementers Forum	PCI-SIG	SATA-IO	INCITS TC T10

Typical x86 PC I/O System



I/O Management

- I/O is mediated by the OS
  - Multiple programs share I/O resources
    - Need protection and scheduling
  - I/O causes asynchronous interrupts
    - Same mechanism as exceptions
  - OS provides I/O abstractions directly to programs
  - Synchronous I/O programming model:
    - Request I/O (e.g., read file from disk, mouse movement, etc.)
    - Kernel issues I/O device specific commands on application's behalf
    - Kernel then returns control to application

I/O Commands at OS level

- I/O devices are managed by I/O controller hardware
  - Transfers data to/from device
  - OS synchronizes operations with software
- OS sets command registers
  - Cause device to do something
- OS queries reserved status registers
  - Indicate what the device is doing and any errors
- OS queries reserved data registers
  - Write: transfer data to a device
  - Read: transfer data from a device

I/O Register Mapping

- Memory mapped I/O
  - Reserve a special part of memory for I/O
  - OS read/writes registers and special memory addresses
  - Hardware address decoder distinguishes between application vs I/O memory commands
  - OS uses address translation mechanism to make them only accessible to OS
- I/O instructions
  - Separate instructions to access I/O registers
  - Can only be executed in kernel mode
  - Example: x86

Polling

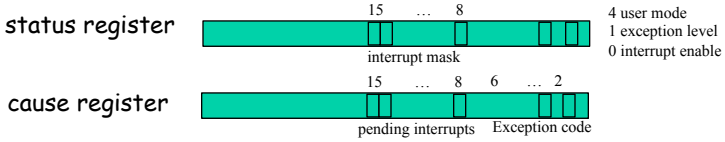
- Hardware periodically checks I/O status register
  - For example, at the beginning of every instruction issue
  - If device ready, do operation
  - If error, take action
- Common in small or low-performance real-time embedded systems
  - Predictable timing
  - Low hardware cost
- In other systems, wastes CPU time

Interrupts

- When a device is ready or error occurs
  - Controller interrupts CPU
- Interrupt is like an exception
  - But not synchronized to instruction execution
  - stops the current program
  - Invokes **OS software handler** between instructions
  - Cause identifies the interrupting device
  - restarts the program or a different one
- Priority interrupts
  - Devices needing more urgent attention get higher priority
  - Can interrupt handler for a lower priority interrupt

I/O Data Transfer

- OS sets status and cause registers



- Polling and interrupt-driven I/O
  - CPU transfers data between memory and I/O data registers
  - Time consuming for high-speed devices
- Direct memory access (DMA)
  - OS provides starting address in memory
  - I/O controller transfers to/from memory autonomously
  - Controller interrupts CPU on completion or error

DMA/Cache Interaction

- If DMA writes to a memory block that is cached
  - Cached copy becomes stale
- If write-back cache has dirty block, and DMA reads memory block
  - Reads stale data
- Need to ensure cache coherence
  - Flush blocks from cache if they will be used for DMA
  - Or use non-cacheable memory locations for I/O

DMA/VM Interaction

- OS uses virtual addresses for memory
  - DMA blocks may not be contiguous in physical memory
- Should DMA use virtual addresses?
  - requires controller to do translation
- If DMA uses physical addresses
  - Must break big transfers into page-sized chunks
  - Or allocate contiguous physical pages for DMA

Measuring I/O Performance

- I/O performance depends on
  - Hardware: CPU, memory, controllers, buses
  - Software: operating system, database management system, application
  - Workload: request rates and patterns
- I/O system design can trade-off between response time and throughput
  - Measurements of throughput often done with constrained response-time

Transaction Processing Benchmarks

- Transactions
  - Small data accesses to a DBMS
  - Interested in I/O rate, not data rate
- Measure throughput
  - Subject to response time limits and failure handling
  - ACID (Atomicity, Consistency, Isolation, Durability)
  - Overall cost per transaction
- Transaction Processing Council (TPC) benchmarks (www.tcp.org)
  - TPC-APP: B2B application server and web services
  - TCP-C: on-line order entry environment
  - TCP-E: on-line transaction processing for brokerage firm
  - TPC-H: decision support — business oriented ad-hoc queries

File System & Web Benchmarks

- SPEC System File System (SFS)
  - Synthetic workload for NFS server, based on monitoring real systems
  - Results
    - Throughput (operations/sec)
    - Response time (average ms/operation)
- SPEC Web Server benchmark
  - Measures simultaneous user sessions, subject to required throughput/session
  - Three workloads: Banking, Ecommerce, and Support

I/O vs. CPU Performance

- Amdahl's Law
  - Don't neglect I/O performance as parallelism increases
- Example
  - Benchmark takes 90s CPU time, 10s I/O time
  - Double the number of CPUs/2 years
    - I/O unchanged

Year	CPU time	I/O time	Elapsed time	% I/O time
now	90s	10s	100s	10%
+2	45s	10s	55s	18%
+4	23s	10s	33s	31%
+6	11s	10s	21s	47%

Summary

- Interconnects & I/O
  - Buses based on I/O device
  - Future networks on chip
  - OS handles I/O
    - with hardware support for interrupts
- Next Time
  - Coarse grain parallelism
  - Parallel programs
  - Shared memory parallel architectures
  - Hardware multithreading
- Reading: P&H 7.1-5