New-School Machine Structures

**Big Idea: Memory Hierarchy**

- **Software**
  - Parallel Requests
    - Assigned to computer
    - e.g., Search "Katz"
  - Parallel Threads
    - Assigned to core
    - e.g., Lookup, Ads
  - Parallel Instructions
    - >1 instruction @ one time
    - e.g., 5 pipelined instructions
  - Parallel Data
    - >1 data item @ one time
    - e.g., Add of 4 pairs of words
  - Hardware descriptions
    - All gates @ one time

- **Hardware**
  - Warehouse Scale Computer
  - Virtual Memory
  - Core
  - [Cache]
  - Instruction Unit(s)
  - Functional Unit(s)
  - Main Memory
  - Logic Gates

Today’s Lecture

New-School Machine Structures

**Big Idea: Memory Hierarchy**

- **Software**
  - Parallel Requests
    - Assigned to computer
    - e.g., Search "Katz"
  - Parallel Threads
    - Assigned to core
    - e.g., Lookup, Ads
  - Parallel Instructions
    - >1 instruction @ one time
    - e.g., 5 pipelined instructions
  - Parallel Data
    - >1 data item @ one time
    - e.g., Add of 4 pairs of words
  - Hardware descriptions
    - All gates @ one time

- **Hardware**
  - Warehouse Scale Computer
  - Virtual Memory
  - Core
  - [Cache]
  - Instruction Unit(s)
  - Functional Unit(s)
  - Main Memory
  - Logic Gates

Today’s Lecture

**Agenda**

- Virtual Memory Revisted
- Administrivia
- Demand Paging
- Exceptions, Traps, Interrupts
- Technology Break
- Virtual Machines
- Summary

**Agenda**

- Virtual Memory Revisted
- Administrivia
- Demand Paging
- Exceptions, Traps, Interrupts
- Technology Break
- Virtual Machines
- Summary
Protection + Indirection = Controlled Sharing

Day in the Life of an (Instruction) Address

Day in the Life of an (Instruction) Address

Day in the Life of an (Instruction) Address

Day in the Life of an (Instruction) Address

No Cache, Virtual Memory, TLB Hit—Very Fast!
If locality works, this is the most common case!

No Cache, Virtual Memory, TLB Miss

No Cache, Virtual Memory, TLB Miss, Page Table Access

NOTE: Virtual Memory implemented before caches

Physical Data Cache, Virtual Memory, TLB Miss, Page Table Access
VA caches are possible, but it's complicated: see CS 362

Physical Data Cache, Virtual Memory, TLB Miss, Page Table Access
VA caches are possible, but it's complicated: see CS 362
**Day in the Life of an (Instruction) Address**

- PC
- PA (VPN, Offset)
- TLB
- Page Table Base Register
- Miss (VPN, Offset)
- Miss
- PA
- DS
- Miss
- PA (VPN, Offset)
- IS

Physical Data & Instruction Cache, Virtual Memory, TLB Miss, Page Table Access

VPN caches are possible, but it’s complicated: see CS 162

---

**Agenda**

- Virtual Memory Revisted
- Administrivia
- Demand Paging
- Exceptions, Traps, Interrupts
- Technology Break
- Virtual Machines
- Summary

---

**Administrivia**

- Extra Credit due 4/24 – Fastest Matrix Multiply
- F2F “Grading” of Project 4 in Lab this week
- Final Review: Mon 5/2, 5 – 8PM, 2050 VLSB
- Final Exam: Mon 5/9, 11:30-2:30PM, 100 Haas Pavilion
  - Designed for 90 minutes, you will have 3 hours
  - Comprehensive (particularly problem areas on midterm), but focused on course since midterm: lecture, lab, hws, and projects are fair game
  - 8 ½ inch x 11 inch crib sheet like midterm

---

**CS61c in the News!**

- NVIDIA Tegra 2 Processor with 1GHz Dual-core ARM Processor
- 1080p MPEG-4/H.264 Recording and Playback
- HDMI mirroring
- 4-inch WVGA screen
- 8-megapixel rear camera / 1.3-megapixel front camera
- 7.1 multi-channel virtual surround sound
- 8GB memory
- microSD memory expandability (up to 32GB)
- Micro-USB connectivity
- 1,500 mAh battery
- Supports Adobe Flash Player 10.1

---

**President Obama @ FB Yesterday!**

CIA’s ‘Facebook’ Program Dramatically Cuts Agency’s Costs

LG Optimus 2X Smart Phone

Theater-quality entertainment on a mobile device

“It’s Game Over for Single-core Smartphones.”
Agenda

- Virtual Memory Revisted
- Administrivia
- Demand Paging
- Exceptions, Traps, Interrupts
- Technology Break
- Virtual Machines
- Summary

Historical Retrospective: 1960 versus 2010

- Memory used to be very expensive, and amount available to the processor was highly limited
  - Now memory is cheap: approx $20 per Gbyte in April 2011
- Many apps’ data could not fit in main memory, e.g., payroll
  - Paged memory system reduced fragmentation but still required whole program to be resident in the main memory
  - For good performance, buy enough memory to hold your apps
- Programmers no longer need to worry about this level of detail anymore

Demand Paging in Atlas (1962)

“A page from secondary storage is brought into the primary storage whenever it is (implicitly) demanded by the processor.”
Tom Kilburn
Primary memory as a cache for secondary memory
User sees 32 x 6 x 512 words of storage

Demand Paging Scheme

- On a page fault:
  - Input transfer into a free page is initiated
  - If no free page available, a page is selected to be replaced (based on usage)
  - Replaced page is written on the disk
    - To minimize disk latency effect, the first empty page on the disk was selected
  - Page table is updated to point to the new location of the page on the disk

Impact on TLB

- Keep track of whether page needs to be written back to disk if it has been modified
- Set “Page Dirty Bit” in TLB when any data in page is written
- When TLB entry replaced, corresponding Page Dirty Bit is set in Page Table Entry

Address Translation: Putting it all Together
Address Translation in CPU Pipeline

- TLB miss? Page Fault?
- Protection violation?

- Software handlers need restartable exception on TLB fault
- Handling a TLB miss needs a hardware or software mechanism to refill TLB
- Need mechanisms to cope with the additional latency of a TLB:
  - Slow down the clock
  - Pipeline the TLB and cache access
  - Virtual address caches (indexed with virtual addresses)
  - Parallel TLB/cache access

Impact of Paging on AMAT

- Memory Parameters:
  - L1 cache hit = 1 clock cycles, hit 95% of accesses
  - L2 cache hit = 10 clock cycles, hit 60% of L1 misses
  - DRAM = 200 clock cycles (~100 nanoseconds)
  - Disk = 20,000,000 clock cycles (~10 milliseconds)

- Average Memory Access Time (with no paging):
  - 1 + 5%*10 + 5%*40%*200 = 5.5 clock cycles

- Average Memory Access Time (with paging):
  - AMAT (with no paging) + ?
  - 5.5 + ?

Impact of Paging on AMAT

- Memory Parameters:
  - L1 cache hit = 1 clock cycles, hit 95% of accesses
  - L2 cache hit = 10 clock cycles, hit 60% of L1 misses
  - DRAM = 200 clock cycles (~100 nanoseconds)
  - Disk = 20,000,000 clock cycles (~10 milliseconds)

- Average Memory Access Time (with paging):
  - 5.5 + 5%*40%*(1-HitMemory)*20,000,000

- AMAT if HitMemory = 99.9999%
  - 5.5 + 0.02 * .001 * 20,000,000 = 405.5

- AMAT if HitMemory = 99.99999%
  - 5.5 + 0.02 * .000001 * 20,000,000 = 5.9

Concurrent Access to TLB & Cache

- TLB Inst.
- TLB miss?
- Page Fault?
- Protection violation?

- Index L is available without consulting the TLB
- cache and TLB accesses can begin simultaneously
- Tag comparison is made after both accesses are completed

Cases: L + b = k, L + b < k, L + b > k

Agenda

- Virtual Memory Revisted
- Administrivia
- Demand Paging
- Exceptions, Traps, Interrupts
- Technology Break
- Virtual Machines
- Summary

Exceptions and Interrupts

- “Unexpected” events requiring change in flow of control
  - Different ISAs use the terms differently

- Exception
  - Arises within the CPU
    - e.g., Undefined opcode, overflow, syscall, ...

- Interrupt
  - From an external I/O controller
  - Dealing with them without sacrificing performance is difficult
Handling Exceptions

- In MIPS, exceptions managed by a System Control Coprocessor (CP0)
- Save PC of offending (or interrupted) instruction
  - In MIPS: save in special register called Exception Program Counter (EPC)
- Save indication of the problem
  - In MIPS: saved in special register called Cause register
  - We’ll assume 1-bit
    - 0 for undefined opcode, 1 for overflow
- Jump to exception handler code at address 8000 0180

Exception Properties

- Restartable exceptions
  - Pipeline can flush the instruction
  - Handler executes, then returns to the instruction
    - Refetched and executed from scratch
- PC saved in EPC register
  - Identifies causing instruction
  - Actually PC + 4 is saved because of pipelined implementation
    - Handler must adjust PC to get right address

Handler Actions

- Read Cause register, and transfer to relevant handler
- Determine action required
- If restartable exception
  - Take corrective action
  - use EPC to return to program
- Otherwise
  - Terminate program
  - Report error using EPC, cause, ...

Exceptions in a Pipeline

- Another kind of control hazard
- Consider overflow on add in EX stage
  - add $1, $2, $1
    - Prevent $1 from being clobbered
    - Complete previous instructions
    - Flush add and subsequent instructions
    - Set Cause and EPC register values
    - Transfer control to handler
- Similar to mispredicted branch
  - Use much of the same hardware

Exception Example

- Exception on add in
  40 sub $11, $2, $4
  44 and $12, $2, $5
  48 or $13, $2, $6
  4C add $1, $2, $1
  50 slt $15, $6, $7
  54 lw $16, 50($7)
  58 lui $14, 1000

- Handler
  8000180 sw $25, 1000($0)
  8000184 sw $26, 1004($0)

Exception Example

- Time (clock cycles)
- Exception Example
- Instruction
  - and
  - or
  - add
  - slt
  - lw
  - lui

Diagram
Imprecise Exceptions

- Just stop pipeline and save state
  - Including exception cause(s)
- Let the software handler work out
  - Which instruction(s) had exceptions
  - Which to complete or flush
    - May require "manual" completion
- Simplifies hardware, but more complex handler software
- Not feasible for complex multiple-issue out-of-order pipelines to always get exact instruction
- All computers today offer precise exceptions—affects performance though

Agenda

- Virtual Memory Revisted
- Administrivia
- Demand Paging
- Exceptions, Traps, Interrupts
- Technology Break
- Virtual Machines
- Summary
Beyond Virtual Memory

- Even greater protection than virtual memory
  - E.g., Amazon Web Services allows independent tasks run on same computer
- Can a "small" operating system simulate the hardware of some machine, so that
  - Another operating system can run in that simulated hardware?
  - More than one instance of that operating system run on the same hardware at the same time?
  - More than one different operating system can share the same hardware at the same time?
- Answer: Yes

Solution – Virtual Machine

- A virtual machine provides interface identical to underlying bare hardware
  - i.e., all devices, interrupts, memory, page tables, etc.
- Virtualization has some performance impact
  - Feasible with modern high-performance computers
- Examples
  - IBM VM/370 (1970s technology!)
  - VMWare
  - Xen (used by AWS)
  - Microsoft Virtual PC

Randy’s Personal Experience
VM/370, circa 1973

- Summer internship @ CoNY Dept Welfare Service
- VM/370: allows programmer to write channel programs, basically machine instructions (CCW —channel control words) to directly control I/O devices
- Let’s try to ring the computer’s console bell
- Terminal log prints out the following:
  !!!!!RRRR....RING....GGGG!!!!!
Virtual Machine Instruction Set Support

- Similar to what need for Virtual Memory
- User and System modes
- Privileged instructions only available in system mode
- Trap to system if executed in user mode
- All physical resources only accessible using privileged instructions
  - Including page tables, interrupt controls, I/O registers
- Renaissance of virtualization support
  - Current ISAs (e.g., x86) adapting, following IBM’s path

And in Conclusion, ...

- Virtual Memory, Paging really used for Protection, Translation, Some OS optimizations
  - Not really routine paging to disk
  - Can think of as another level of memory hierarchy, but not really used like caches
- Virtual Machines as even greater level of protection to allow greater level of sharing
  - Enables fine control, allocation, pricing of Cloud Computing

Peer Instruction: Match the Phrase

Match the memory hierarchy element on the left with the closest phrase on the right:

1. L1 cache  a. A cache for page table entries
2. L2 cache  b. A cache for a main memory
3. Main memory c. A cache for disks
4. TLB       d. A cache for a cache

A] 1 a, 2 b, 3 c, 4 d  
B] 1 a, 2 b, 3 d, 4 c  
C] 1 b, 2 d, 3 a, 4 c  
D] 1 d, 2 b, 3 c, 4 a

Peer Instruction: True or False

A program tries to load a word X that causes a TLB miss but not a page fault. Which are True or False:

1. A TLB miss means that the page table does not contain a valid mapping for virtual page corresponding to the address X
2. There is no need to look up in the page table because there is no page fault
3. The word that the program is trying to load is present in physical memory.

RED] 1 f, 2 f, 3 F  
ORG] 1 f, 2 f, 3 T  
GRN] 1 f, 2 T, 3 F  
YEL] 1 f, 2 T, 3 T

TEAL] 1 d, 2 c, 3 b, 4 a

PNK] 1 T, 2 F, 3 F
PUR] 1 d, 2 a, 3 b, 4 c
BLU] 1 d, 2 b, 3 a, 4 c
GRN] 1 b, 2 d, 3 c, 4 a

Agenda

- Virtual Memory Revisted
- Administrivia
- Demand Paging
- Exceptions, Traps, Interrupts
- Technology Break
- Virtual Machines
- Summary
**Peer Instruction: True or False**

A program tries to load a word X that causes a TLB miss but not a page fault or protection violations. Which are True or False:

1. A TLB miss means that the page table does not contain a valid mapping for virtual page corresponding to the address X
2. There is no need to look up the page table because there is no page fault
3. The word that the program is trying to load is present in physical memory.

A) 1 F, 2 T, 3 F, 4 T  
B) 1 F, 2 F, 3 T  
C) 1 F, 2 T, 3 F  
D) 1 F, 2 T, 3 T

---

**Peer Instruction: True or False**

TLB entries have valid bits and dirty bits. Data caches have them also.

A. The valid bit means the same in both: if valid = 0, it must miss in both TLBs and Caches.
B. The valid bit has different meanings. For caches, it means this entry is valid if the address requested matches the tag. For TLBs, it determines whether there is a page fault (valid=0) or not (valid=1).
C. The dirty bit means the same in both: the data in this block in the TLB or Cache has been changed.
D. The dirty bit has different meanings. For caches, it means the data block has been changed. For TLBs, it means that the page corresponding to this TLB entry has been changed.

A) 1 F, 2 T, 3 F, 4 T  
B) 1 F, 2 T, 3 T, 4 F  
C) 1 T, 2 F, 3 F, 4 T  
D) 1 T, 2 T, 3 T, 4 T