EE 108B Review Session #7

Njuguna Njoroge
Thursday, December 7, 2005
Gates B03
Live on E4, 9:30-10:45 AM

Today's Agenda…

- Announcements
- Final Exam Logistics
- Snap-shot of quarter's topics
- Final Exam preparation tips
- Sample problems (previous finals and other sources – watch review session for solutions)

Announcements

- HW 5 solutions to be posted on Friday (12/9)
- HW 5 will be returned on Thurs (12/15) -- graders have finals, so they can cannot return them sooner.
- No more TA OH (we have finals as well)
- Professor's finals OH TBA

Final Exam Logistics

- URL with info: http://www.stanford.edu/class/ee108b/exams.html
- Date and Time: Wed, 12/14/05, 12:15 PM-3:15 PM
- Location: Gates B03
- Coverage: Lectures 1-19
- Format: open-book and open-notes. No electronic devices other than a calculator will be permitted.
- Local SCPD students: must attend (within a 50-mile radius of Stanford).
  - Parking is difficult to find on campus → Arrive early (45 minutes in advance)
- Remote SCPD students:
  - can take the test any time on the same day.
  - their proctors will receive a PDF copy of the exam from SCPD the morning of the exam. Contact SCPD if there are any issues in the delivery.

Main Topics Covered this Quarter

- MIPS Assembly Language
- CPU Performance and Compiler Optimizations
- Single Cycle Processor Design

pre-midterm

post-midterm

- Pipeline Processor Design
- Memory Design
- Processes, Interrupts and Exceptions
- Virtual Memory
- I/O and OS basics – Bus interfaces, DMA Controllers, etc.
- Intro to Multiprocessor Design

Final Exam Preparation Tips

- Limited amount of study-time, so use time wisely
- My strategy for these kind of classes

- Exam topics are from whole ¼, but there is more emphasize on post-midterm material

The inquisition begins … Short Answer Questions

- What's the difference between bandwidth and latency?
- When is latency important, but BW is unimportant?
- When is BW important, but latency isn't?

Application Locality

- Rank the application classes in terms of instruction and data locality (1 = lowest locality, 2 = middle locality, 3 = highest locality)

Application	Instruction Locality	Data Locality
Desktop (excel, word, etc)		
Scientific (matrix multiply, linpack)		
Database (TPC-C)		

Register-Mem example

- ISAs such as the x86 contain register-memory instructions in which one of the source operands is a memory address. Here is an example:
- `memadd $1, $2, $3 # $1 = Memory[$2] + $3`
- Describe the minimum change that needs to be made to the MIPS pipeline to accommodate this instruction. What impact would this change have on other load and store instructions?

Multiprocessor's performance limits

- Name the two fundamental limiting factors for the performance that a multiprocessor (parallel) system can achieve on an application.

Cache Set Associativity

- Why would you build a 32-way set associative primary cache if the miss rate is no better than an 8-way set associative cache?

Cache Performance

Consider the following loop:

```
struct position {
   int x;
   int y;
}
struct position grid[16][16];
int total_x = 0, total_y = 0;
int x,y;
for (i = 0; i < 16; i++) {
   for (j = 0; j < 16; j++) {
       total_x += grid[i][j].x;
   }
}
// Note that the loops have been interchanged
for (j = 0; j < 16; j++) {
   for (i = 0; i < 16; i++) {
       total_y += grid[i][j].y;
   }
}
```

Cache Problem Cont'd

- The code runs on a machine with the following data cache:
  - direct-mapped
  - total size: 512 bytes
  - block size: 16 bytes
- Additional assumptions:
  - sizeof(int) == 4
  - grid begins at memory address 0
  - Cache is initially empty
  - The only data memory accesses are to the entries of the array grid. Variables `i, j, total_x`, and `total_y` are stored in registers.

- So how many sets?

Cache Performance: Part A

- What is the total number of data reads?

Cache Performance: Part B

- What is the total number of reads that miss in the cache and what type of misses are they?

Cache Performance: Part B cont'd

Cache Performance: Part D

- Why is it safe to ignore the impact of instruction references on cache performance in this problem?

Cache Performance: Part E

- Re-write the code snippet in as few lines as possible to improve cache performance. What type of locality does your improved code increase?

Virtual Cache (VC) and Exceptions

- What's the difference between a physically tagged and virtually tagged cache?

- What is the main advantage of using a virtually tagged cache?

VC and Exceptions cont'd

- Name some disadvantages of a virtually tagged cache.

VC and Exceptions cont'd

- An architecture is said to support precise exceptions if what following conditions are met?

VC and Exceptions cont'd

- For the following types of exceptions, state whether the process should resume execution after the exception and if so, whether execution should resume at the faulting instruction (the instruction that causes the exception) or the instruction after the faulting instruction.

Exception Types	**Resume**	**Re-execute faulting instruction**
Misaligned address		
Page fault		
Memory protection violation		
Syscall		

VC and Exceptions cont'd

- Why is it important to support precise exceptions in an architecture that uses virtual memory?

MIPS ISA and Pipelining

We have decided to extend the MIPS ISA to support a method of memory addressing known as *memory indirect*. In memory indirect addressing, the memory address is dereferenced, as if following a pointer. The instruction load memory indirect (lmi) would look like:

```
lmi $r0, d($r1)        # $r0 <= Mem[Mem[$r1 + d]]
```

The instruction for store memory indirect (smi) would look like:

```
smi $r0, d($r1)        # Mem[Mem[$r1 + d]] <= $r0;
```

The pipeline needs to be changed to accommodate the new instruction. Additional hardware has been added, as well as one new pipe stage.

IF	Instruction fetch and new PC calculation
ID	Instruction decode, register fetch, and branch condition and address calculation
EX	ALU operations with 2 registers as inputs are performed, or load/store effective address calculation.
MEM1	Data memory access is performed
MEM2	Data memory access is performed
WB	Operation results are written back to the register file (the reg file has to wait until the beginning of the next cycle to access this newly written data)

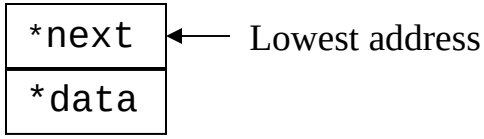
MIPS ISA and Pipelining cont'd

- What resources are necessary to support this pipeline? Specifically, how does the reg file and memory structure look like?

MIPS ISA and Pipelining cont'd

Below we have linked list composed of the elements called ee108b_listnode. In C, the first field of struct, has the lowest address, which would be the *next field in this struct.

```
typdef struct ee108b_listnode {
        struct eel08b_listnode *next;
        int *data;
} ee108b_listnode_t;
```



The snippet of code below operates on a linked list formed of these ee108b_listnodes.

```
# $a0 = address of list header pointer;
# $a1 = non-zero integer
# $vx = return values
            add         $v0, $zero, $zero
            add         $s0, $zero, $zero
            lw          $v1, 0($a0)
            j           test
            nop
loop:
            addi        $v0, $v0, 1
            lw          $t0  4($v1)
            lw          $s0         0($t0)
            lw          $v1, 0($v1)
test:
            beq         $v1,$zero done
            sub         $t0, $s0, $a1
            beq         $t0,$zero done
            nop
            j           loop
            nop
done:
```

- What does this code do? Describe in a few sentences

MIPS ISA and Pipelining cont'd

- Suppose the loop is executed once and exits the loop on the second branch. How many stalls will have to be inserted if the code below is run on the new 6-stage pipeline? Assume that all necessary forwarding paths are implemented. What if we use the new memory indirect instructions? Circle the instructions that can be replaced with the new instructions. How many stalls are needed now? Please show the number of stalls next to wherever the stall occurs and write the total at the bottom.
- Note: You *can only* forward from the *end* of a stage to the *beginning* of the destination stage

MIPS ISA and Pipelining cont'd

Instruction	1	2	3	4	5	6	7	8	9	10	11
i	IF	ID	EX	MEM1	MEM2	WB					
i+1		IF	ID	EX	MEM1	MEM2	WB				
i+2			IF	ID	EX	MEM1	MEM2	WB			
i+3				IF	ID	EX	MEM1	MEM2	WB		
i+4					IF	ID	EX	MEM1	MEM2	WB	
i +5						IF	ID	EX	MEM1	MEM2	WB

	2 iterations of loop	Num stalls (5-stage pipeline)	Num stalls (new pipeline)
	add $v0, $zero, $zero		
	add $s0, $zero, $zero		
	lw $v1, 0($a0)		
	j test		
	nop		
test:	beq $v1,$zero done		
	sub $t0, $s0, $a1		
	beq $t0,$zero done		
	nop		
	j loop		
	nop		
loop:	addi $v0, $v0, 1		
	lw $t0, 4($v1)		
	lw $s0, 0($t0)		
	lw $v1, 0($v1)		
test:	beq $v1,$zero done		
	sub $t0, $s0, $a1		
	beq $t0,$zero done		
	nop		
done:			
	Total stalls →		

MIPS ISA and Pipelining cont'd

- If it were up to you, would you implement this pipeline? In your explanation, cite some of the pros and cons of this new pipeline and which influenced your decision on implementing this pipeline.

That's all folks…

- All the best on the final exam!
- It was a pleasure TA-ing you all this quarter