Universidad Carlos III de Madrid

Computer Architecture

Group 89

Midterm exam

Name:_____

NIA: _____

I. (10 points) Answer the following questions. ***A correct answer scores 1 point, while an incorrect answer subtracts 0.25 points.***

Question	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Answer	**C**	**C**	**B**	**A**	**B**	**B**	**A**	**A**	**C**	**A**

1. An application kernel is: a) an operating system designed for just one application b) a set of instructions of an applications that cause most performance problems c) a key piece of a real application

2. For which of the following classes of computers is the throughput most important a) mobile devices b) desktops c) servers d) embedded computers

3. n-bit predictors are static branch prediction techniques: a) True b) False

4. Loop unrolling is a compile-time technique. a) True b) False

5. When data is in the cache (hit), write operations are faster for write-through than for write back caches. a) True b) False

6. The second level of a two-level page table can never be stored ONLY on disk. a) True b) False

7. Giving priority to reads over writes targets to: a) reduce miss penalty b) reduce miss rate c) increase cache bandwidth d) reduce hit time

8. Using multilevel caches targets to: a) reduce miss penalty b) reduce miss rate c) increase cache bandwidth d) reduce hit time

9. Using multiple bank memories targets to: a) reduce miss penalty b) reduce miss rate c) increase cache bandwidth d) reduce hit time

10. Merging write buffer targets to: a) reduce miss penalty b) reduce miss rate c) increase cache bandwidth d) reduce hit time

II. (20 points) Shortly answer the following questions:

a) Discuss the relationship between branch prediction and hardware prefetching

b) For the 5 stages MIPS pipeline explain why a data cache and an instruction cache may avoid a structural hazard.

c) Compare a fully associative cache and a direct-mapped cache in terms of hit time and cost.

d) Explain how is it possible to parallelize virtual address translation and indexing the cache.

e) Explain "critical word first" cache optimization.

Solution:

a) Branch prediction predicts next instruction when a branch is ecountered, can be used by hardware prefetching for bringing the right block to instruction cache

b) In a pipelined execution the fetch and memory phase can access in the same time the cache.

c) Fully associative cache has higher cost and higher hit time.

d) Use the page offset to index the cache.

e) Bring first the word of a block that is used in the current instruction.

III. (10 points) A program consists of a loop executed 1000 times. Inside the loop the program executes five conditional branches whose outcome is TNNTN. List the predictions and calculate the branch prediction accuracy for the following branch schemes:

a) Always taken
b) Always not-taken
c) 1-bit predictor starting in the state predict not-taken
d) 2 bit predictor starting in the strongly predict non-taken.

Solution 5 branches TNNTN:

a) 40% b) 60% c) 20% d) 60%

Solution 6 branches TNNTNT:

a) 50% b) 50% c) 33% d) 33%

IV. (20 points) In the system we are analyzing the memory has:

• Separate L1 instruction and data caches, HitTime = Processor Cycle Time

• 32KB L1 instruction cache with 2% miss rate, 64B blocks

• 256KB L1 data cache with 5% miss rate, 16B blocks

• 256K L2 unified cache with 64B blocks, local miss rate 20%, Hit Time = 4 cycles,

• Main Memory Access time is 50 cycles for the first 64 bits and subsequent 64 bit chunks are available every 10 cycles.

• Both L1 caches are four-way associative, L2 direct mapped.

• Assume there are no misses to main memory.

(a) What is the Miss Penalty for accesses to L2?

(b) What is the average memory access time for instruction references?

(c) What is the average memory access time for data references?

(d) Assume the only memory reference instructions are loads(25%) and stores(5%). What percentage of total memory references are data references?

(e) What is the Average memory access time?

Solution

(a)

64bits=8B transferred at a time, so there is a total of 64B/8B = 8 chunks transferred

HitTime main−memory = initial chunk + 7 remaining chunks

HitTime main−memory = 50 + 7 ∗ 10 cycles

HitTime main−memory = 120 cycles

(b)

AMAT = Hit TimeL1 + MRL1 ∗ MPL1

AMAT = HitT ime + MRL1 ∗ (HitTimeL2 + MRL2 ∗ MPL2)

AMATInstruction = 1 + MRL1−Instruction ∗ (HitTimeL2 + MRL2 ∗ MPL2)

AMATInstruction = 1 + .02 ∗ (4 + .20 ∗ 120)

AMATInstruction = 1 + .02 ∗ (4 + 24)

AMATInstruction = 1.56

(c)

AMATData = 1 + MRL1−Data ∗ (HitT imeL2 + MRL2 ∗ MPL2)

AMATData = 1 + .05 ∗ (4 + .20 ∗ 120)

AMATData = 1 + .05 ∗ (28)

AMATData = 2.4

(d)

Let #instructions = I then

#datareferences = .3*I

So the %total memory references that are data references is

$.3 * I / (I + .3 * I) = .3 / 1.3 = .23$

(e)

$AMAToverall = \%InstrRef * AMATInstr + \%DataRef * AMATData$

$AMAToverall = (1 - .23) * AMATInstr + .23 * AMATData$

V. (30 points) Loop Unrolling - Given the code

```
$L46:
        addu $2, $2, $3
        l.d $f0, 0($2)
        mul.d $f2, $f0, $f0
        add.d $f4, $f2, $f0
        s.d $f4, 0($4)
        subui $4, $4, #8
        beq $2, $4, $L46
```

(a) Show how to unroll the loop just once.

(b) If your loop would normally execute an odd number of times and your unroll it once, what problems arise and how do you address them.

(c) What in the architecture limits the number of times a loop could be unrolled?

(d) Assuming that you have four pipelines, such that it is possible to pipeline 1 can issue integer operations, pipeline 2 floating add operations, pipeline 3 floating multiply operations and pipeline 4 load/store operation, show how to optimally schedule the unrolled loop.

Solution:

$L46:

1. addu $2, $2, $3
2. l.d $f0, 0($2)
3. l.d $f6, 8($2)
4. mul.d $f2, $f0, $f0
5. mul.d $f8, $f6, $f6
6. add.d $f4, $f2, $f0
7. add.d $f10, $f8, $f6
8. s.d $f4, 0($4)
9. s.d $f10, 0($4)
10. subui $4, $4, #16
11. beq $2, $4, $L46

(b) Handle the work done normally in the last iteration outside the loop.

(c) The number of registers.

(d)

iadd	f.add	f.mul	L/s

1			
			2
		4	3
	6	5	
	7		8
			9
10			
11			

VI. (10 points) Assume a classic 5-stage pipeline that has been extended to allow multiple cycle operations, with the execution stage of FP multiplies taking 4 cycles, loads and stores taking 1 cycle, FP adds take 2 cycles and integer operations take 1 cycle.

a) Identify all the hazards.

b) Show how the code below moves through the pipeline. Stop with cycle 17.

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
addu $2,$2,$3	F	D	E	M	W														
l.d $f0,0($2)																			
mul.d $f2,$f2,$f0																			
l.d $f8,1000($2)																			
mul.d $f4,$f4,$f8																			
add.d $f6,$f2,$f4																			
s.d $f6,0($4)																			

Solution

a) Data dependencies that can result in hazards: 1-2, 2-3, 1-4, 4-5,3-6. 5-6, 6-7

b) Solution with no forwarding (solution with forwarding also accepted)

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
addu $2,$2,$3	F	D	E	M	W														
l.d $f0,0($2)		F			D	E	M	W											
mul.d $f2,$f2,$f0					F			D	E				M	W					
l.d $f8,1000($2)								F	D				E	M	W				
mul.d $f4,$f4,$f8									F						D	E			
add.d $f6,$f2,$f4															F	D			
s.d $f6,0($4)																F			