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	D	C	A	B	B	A	C	A	A

1. A benchmark is: a) The maximum performance a computer can achieve b) A grade the performance of a computer is assigned c) A program that assesses the performance of a computer.

2. The speedup is NOT a) The performance gain when executing on a faster architecture. b) The performance gain when applying an enhancement to an application. c) The performance gain when running on larger number of nodes d) The superior speed of an application over another application.

3. A structural hazard arises: a) When an instruction depends on the result of a previous instruction b) From pipelining of branches that change the program counter (PC) c) From resource conflicts, when hardware cannot support simultaneously all instructions in overlapped execution

4. A control dependency does not allow for static scheduling. a) True b) False

5. Loop unrolling is a run-time technique. a) True b) False

6. Reducing the miss rate reduces the miss penalty a) True b) False

7. Write-allocate can be used both for write-through and write-back caches. a) True b) False

8. Which cache structure is the most simple to implement (less hardware)? a) associative b) 2-way set associative c) direct-mapped

9. A page table may contain the mapping of a virtual page to a disk block. a) True b) False

10. A virtual machine has a virtual memory on top of the virtual memory of a physical machine a) True b)

False

II. (20 points) Shortly define the following concepts:

a) Amdahl's law b) embedded computers c) dynamic scheduling d) branch prediction e) conflict misses

f) write-allocate

III. (10 points) For the following code:

1: MOV R3, R7
2: LD R8, (R3)
3: ADD R3, R3, 4
4: LOAD R9, (R3)
5: BNE R8, R9, L3

a) Identify WAW, RAW, and WAR dependencies.
b) What is the difference between a dependency and hazard?
c) What is the difference between a name dependency and a true dependency?
d) Which of WAW, RAW, WAR are true dependencies and which are name dependencies?

Solution:

a)

WAW: 1-3

RAW: 1-2, 1-3, 3-4, 2-5, 4-5

WAR: 2-3

b) A hazard occurs when there is a dependence between instructions and the instructions are closed enough that their execution would result in violating the dependence. i.e. , not all dependences result necessarily in hazards.

c+d) For a true dependence there is a flow of data from the first instruction to the second instruction (RAW). For a name dependence there is no flow of data and the two instructions just use the same register or memory location, which can be replaced for another one in order to eliminate the dependence (WAW, WAR).

IV. (30 points) Given the following code:

```
1: Loop:  LD F2,0(Rx)
2:        MULTD F2,F0,F2
3:        DIVD F8,F2,F0
4.        LD F4,0(Ry)
5:        ADDD F4,F0,F4
6:        ADDD F10,F8,F2
7:        SD F4,0(Ry)
8:        ADDI Rx,Rx,#8
9:        ADDI Ry,Ry,#8
10:       SUB R20,R4,Rx
11:       BNZ R20,Loop
```

and the following latencies for the instructions:

Instruction	LD	SD	Integer ADD, SUB	Branches	ADDD	MULTD	DIVD
Latency	4	2	1	2	3	5	11

Answer the following questions:

a) Assuming that: one instruction per cycle is issued, one pipeline is available, the execution is in-order and the branch is taken. How many cycles per iterations are needed to execute one loop? Ignore in your calculations the initial fetch and decode.
b) Identify the true data dependencies in the code.
c) Assume that: you have two execution pipelines, each executing one instruction per cycle, you have enough fetch/decode bandwidth and no instruction reordering is performed. How many cycles does the loop require?
d) Reorder the instructions to improve performance (data dependencies and functional unit latencies must be observed. How many cycles does the loop require? What is the speedup over one pipeline and two pipelines without reordering?

Solution

a)

```
Loop: LD F2,0(Rx)
        <stall>
        <stall>
        <stall>
        MULTD F2,F0,F2
        <stall>
        <stall>
        <stall>
        <stall>
        DIVD F8,F2,F0
        LD F4,0(Ry)
        <stall due to LD latency>
        <stall due to LD latency>
        <stall due to LD latency>
        ADDD F4,F0,F4
        <stall due to DIVD latency>
        <stall due to DIVD latency>
        <stall due to DIVD latency>
        <stall due to DIVD latency>
```

<stall due to DIVD latency>
ADDD F10,F8,F2
SD F4,0(Ry)
ADDI Rx,Rx,#8
ADDI Ry,Ry,#8
SUB R20,R4,Rx
BNZ R20,Loop
<stall due to BNZ>
———

27 cycles per loop

b) 1-2, 2-3, 3-6, 4-5, 2-6, 5-7, 8-10, 10-11

c)

Execution pipe 0	Execution pipe 1
Loop: LD F2,0(Rx)	; <nop>
<stall for LD latency>	; <nop>
<stall for LD latency>	; <nop>
<stall for LD latency>	; <nop>
MULTD F2,F0,F2	; <nop>
<stall for MULTD latency>	; <nop>
<stall for MULTD latency>	; <nop>
<stall for MULTD latency>	; <nop>
<stall for MULTD latency>	; <nop>
DIVD F8,F2,F0	; LD F4,0(Ry)
<stall for LD latency>	; <nop>
<stall for LD latency>	; <nop>
<stall for LD latency>	; <nop>
ADDD F4,F0,F4	; <nop>
<stall due to DIVD latency>	; <nop>
<stall due to DIVD latency>	; <nop>
<stall due to DIVD latency>	; <nop>
<stall due to DIVD latency>	; <nop>
<stall due to DIVD latency>	; <nop>
<stall due to DIVD latency>	; <nop>
ADDD F10,F8,F2	; SD F4,0(Ry)
ADDI Rx,Rx,#8	; ADDI Ry,Ry,#8
SUB R20,R4,Rx	; <nop>
BNZ R20,Loop	; <nop>
<stall due to BNZ>	; <nop>

25 cycles per loop

d)

Execution pipe 0	Execution pipe 1
Loop: LD F2,0(Rx)	; LD F4,0(Ry)
<stall for LD latency>	; <stall for LD latency>
<stall for LD latency>	; <stall for LD latency>
<stall for LD latency>	; <stall for LD latency>
MULTD F2,F0,F2	; ADDD F4,F0,F4
<stall for MULTD latency>	; <stall for ADDD latency>
<stall for MULTD latency>	; <stall for ADDD latency>
<stall for MULTD latency>	; SD F4,0(Ry)

```
<stall for MULTD latency>     ; <nop>
DIVD F8,F2,F0                 ; <nop>
<stall for DIVD latency>      ; <nop>
<stall for DIVD latency>      ; <nop>
<stall for DIVD latency>      ; <nop>
<stall for DIVD latency>      ; <nop>
<stall for DIVD latency>      ; <nop>
<stall for DIVD latency>      ; <nop>
<stall for DIVD latency>      ; <nop>
<stall for DIVD latency>      ; <nop>
ADDD F10,F8,F2                ; ADDI Ry,Ry,#8
SUB R20,R4,Rx                 ; <nop>
ADDI Rx,Rx,#8                 ; BNZ R20,Loop
<stall due to BNZ>            ; <stall due to BNZ>
```

22 cycles per loop

Speedup over one pipeline = 27/22

Speedup over two pipelines no reordering = 25/22

V. (20 points) Given a ***fully associative*** 128-byte instruction cache with a 4-byte block (every block can hold exactly one instruction). The cache uses an LRU replacement policy.

a) What is the instruction miss rate for a loop with a very large number of iteration if the loop size is (i) 64-byte (ii) 192 bytes (iii) 320 bytes

b) If the replacement policy is changed to MRU (most recently used), which of the three cases from above would benefit?

c) For the cases in which the miss if high, propose an optimization of the loop that would improve the performance.

Solution:

a) The 64-byte loop will completely fit in the 128 byte cache and the miss rate will be 0%. 192-byte and 320-byte loops will produce a 100% miss rate.

b) The 64-byte loop does not have any misses and will not benefit from MRU. The 192-bytes will have 31 hits and 17 misses in every iteration. The 320-byte loop will have 31 hits and 49 misses in every iteration.

c) Break the loop into smaller loops having at most 64 bytes.

VI. (10 points) Increasing a cache's associativity (with all other parameters kept constant) statistically reduces the miss rate. Given a cache with a total size of 4 bytes, having 4 entries, you are asked to.

a) Construct a sequence of memory accesses (addresses) which provides a worse miss rate for a 2-way associative cache than a direct-mapped cache. Which is the hit rate? (The sequence must be in the form 4,7,8,3, 2 ,4 …, where the numbers represent memory addresses.) Assume the set-associative cache uses a LRU replacement policy.

b) Construct a sequence of memory accesses (addresses) which provides a worse miss rate for a fully associative cache than a two-way associative cache. Which is the hit rate? Assume both caches use a LRU replacement policy.

Solution:

M: miss, H:hit

a) 0,2,4,0,2,4,0,2,4 … will produce for

direct-mapped: M, M, M, M, H, M, M, H, M, …. , i.e. 33% hit rate

two-way cache: M, M, M, M, M, M, M, M, M, …., i.e. 0% hit rate

b) 0,1,2,3,4,0,1,2,3,4,0,1,2,3,4 … will produce

fully-associative: M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, ... i.e. 0% hit rate

two-way cache: M, M, M, M, M, M, H, M, H, M, M, H, M, H, M.... i.e 40% hit rate