CSE 490/590 Computer Architecture
Midterm Solution

DIRECTIONS

- Time limit: 45 minutes (12pm - 12:45pm)
- There are 40 points plus 5 bonus points.
- This is a closed-book, no calculator, closed-notes exam.
- Each problem starts on a new page.
- Please use a pen, not a pencil. If you use a pencil, it won't be considered for regrading.
- Each problem also explains its grading criteria. "Atomic" means that you get either all the points or no point, i.e., there are no partial points.

Name:	
UBITName:	

Problem #	Score
1	
2	
3	
4	
5	
6	

1. (a) List and explain three types of pipeline hazards.
(**Grading**: total 3 points; 1 point for each type; no point without a correct explanation)
Answer:
Structural hazards: an instruction in the pipeline may need a resource being used by another instruction in the pipeline.

Data hazards: an instruction may depend on a data value produced by an earlier instruction.

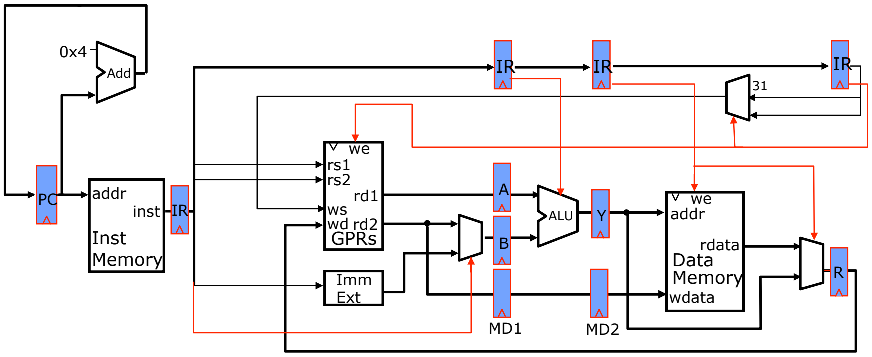
Control hazards: an instruction may depend on the next instruction's address produced by an earlier instruction.

(b) List and explain three types of cache misses.
(**Grading**: total 3 points; 1 point for each type; no point without a correct explanation)
Answer:
Compulsory: first-reference to a block a.k.a. cold start misses. These are the misses that would occur even with infinite cache.

Capacity: cache is too small to hold all data needed by the program. These are the misses that would occur even under perfect replacement policy & full associativity.

Conflict: misses that occur because of collisions due to block-placement strategy. These are the misses that would not occur with full associativity.

2. Suppose that you have the following pipelined datapath with two characteristics — (i) it always speculates that it will execute PC + 4 next and (ii) the register file (i.e., GPRs) allows writing to a register and reading the written value from the same register in the same one cycle.



Also suppose that you have executed the following sequence of instructions with the above datapath, where “ADD R1, R2, R3” adds two registers R2 and R3 and stores the result to R1 (i.e., R1 $\leftarrow$ R2 + R3), and “SUB R1, R2, R3” substracts R3 from R2 and stores the result to R1 (i.e., R1 $\leftarrow$ R2 $-$ R3).

Instruction Address	Instruction
100	ADD R3, R1, R4
104	SUB R4, R3, R1
108	SUB R5, R3, R2
112	ADD R2, R5, R3
116	ADD R1, R4, R5

What is the final value for each of R1, R2, and R5? Use the following initial values.
(**Grading:** total 6 points; 2 atomic points for each correct register)

Answer:

R1	30	30	30	30	30	30	30	30	90
R2	40	40	40	40	40	40	40	160	160
R3	50	50	50	50	90	90	90	90	90
R4	60	60	60	60	60	20	20	20	20
R5	70	70	70	70	70	70	10	10	10
ADD R3, R1, R4	IF	ID	EX	MA	WB				
SUB R4, R3, R1		IF	ID	EX	MA	WB			
SUB R5, R3, R2			IF	ID	EX	MA	WB		
ADD R2, R5, R3				IF	ID	EX	MA	WB	
ADD R1, R4, R5					IF	ID	EX	MA	WB

3. Suppose that you have a cache with the following characteristics.

Word size	32 bits
Cache block size	4 words
Total cache size	128 bytes
Access unit	Word

Suppose further that (i) memory addresses are byte-granularity, (ii) initially the cache is empty, and (iii) you have the following sequence of memory read accesses.

Read Address Sequence
144
25
146
409
20

(a) Assume that the cache is 2-way associative and the cache replacement policy is LRU. Fill in the index and indicate whether each read is a cache hit or a miss in the following table.
(**Grading:** total 5 points; 1 point for each correct cache access)

Read Address Sequence	Address in Binary	Cache Index Value For the Read	Hit or Miss
144	0b000010010000	0b01	Miss
25	0b000000011001	0b00	Miss
146	0b000010010010	0b00	Hit
409	0b000110011001	0b01	Miss
20	0b000000010100	0b01	Miss

Answer:
Please refer to the table.

(b) With the same read sequence, calculate the miss rate if the cache is direct-mapped.
(**Grading:** atomic 2 points)
Answer:
100%

4. Consider the following two code snippets that perform the same computation.

   Code 1:

```
for (j = 0; j < 4; j++)
    for (i = 0; i < 4; i++)
        x[i][j] = 10 * y[i][j];
```

   Code 2:

```
for (i = 0; i < 4; i++)
    for (j = 0; j < 4; j++)
        x[i][j] = 10 * y[i][j];
```

   Suppose that your cache has the following characteristics. Assume that (i) the cache is empty when you start the execution of each of the above code snippets, and (ii) the cache is used only for the matrices, X and Y.

Cache size	16 matrix elements (e.g., one element can be $Y[0][0]$)
Associativity	Fully-associative
Replacement policy	LRU
Write policy	Write-back and write-allocate

   (a) If the cache block size is 4 matrix elements, will you run code 1 or code 2? Explain why. (**Grading:** 5 points)

   Answer:

   Code 2. Better spatial locality.

   (b) If the cache block size is 2 matrix elements, will you run code 1 or code 2? Explain why. (**Grading:** 5 points)

   Answer:

   Either code 1 or code 2. No improvement at all.

   (c) Determine the minimum cache block size (in terms of matrix elements) that makes both code snippets perform equally in terms of cache miss rate.
   (**Grading:** atomic 2 points)

   Answer:

   1 matrix element — every access is a miss.

5. What is the average memory access time when you have the following memory hierarchy? Assume that (i) the cache uses physical addresses, (ii) the CPU stalls until the data is delivered, (iii) everything fits into the memory, and (iv) the hardware does the page table walk and updates TLB. (**Grading:** atomic 4 points)

Unit	Additional Access Latency	Local Hit rate
TLB	1 cycle	95%
L1	1 cycle	95%
L2	8 cycles	80%
L3	50 cycles	50%
Memory	100 cycles	100%
Page table walk & TLB update	200 cycles	100%

Answer: 13.02

Equation:

(Average memory access time) = (TLB access) $+\frac{95}{100}\times$ (L1 hit time) $+\frac{5}{100}\times$ (L1 miss penalty)

(TLB access) = (TLB hit time) $\times\frac{95}{100}+$ $\{$(TLB hit time) $+$ (TLB miss penalty)$\}$ $\times\frac{5}{100}$

(L1 miss penalty) = $\frac{80}{100}\times$ (L2 hit time) $+\frac{20}{100}\times$ (L2 miss penalty)

(L2 miss penalty) = $\frac{50}{100}\times$ (L3 hit time) $+\frac{50}{100}\times$ (L3 miss penalty)

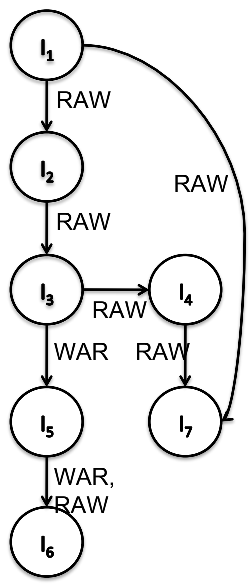
(L3 miss penalty) = (memory hit time)

6. Consider the following sequence of instructions.

Instruction Number	Instruction
I_1	SUBD F4, F4, F0
I_2	DIVD F3, F4, F0
I_3	ADDD F6, F3, F2
I_4	MULTD F1, F6, F7
I_5	ADDD F2, F5, F9
I_6	DIVD F5, F9, F2
I_7	ADDD F8, F1, F4

(a) Draw the dependency graph among the instructions and indicate the type of data hazards for each edge. (**Grading:** atomic 4 points)

Answer:



(b) Assuming that you have as many functional units as you need, list all possible valid instruction sequences. (**Grading:** total 6 points; 1 point for each correct sequence)

Answer:

1, 2, 3, 4, 5, 6, 7
1, 2, 3, 4, 5, 7, 6
1, 2, 3, 4, 7, 5, 6
1, 2, 3, 5, 4, 6, 7
1, 2, 3, 5, 6, 4, 7
1, 2, 3, 5, 4, 7, 6