CSE4010-01: Computer Architecture and Logic
School of Computer Science & Eng., Sogang University

CSE4010-01 Computer Architecture and Logic
Final Exam

- **Exam instructions**

Answer each of the questions included in the exam. Write all of your answers directly on the examination paper, including any work that you wish to be considered for partial credit. The examination is closed-book, and you may use a calculator. You cannot use any other electronic devices including your cellular phone. You will have 75 minutes to complete this exam. Budget your time and try to leave some time at the end to go over your work. You can handout your exam and leave the classroom if you complete the exam before the due time.

Name: _____

ID: _____

		Score
1. General questions	**(10)**	_____
2. Pipelined processor	**(25)**	_____
3. Code Evaluation for Cache	**(30)**	_____
4. Cache Performance	**(15)**	_____
Total	**(80)**	_____

1. Please fill the blank with O if the statement is true. Otherwise fill it with X.

a. Pipelined CPU increases CPI over unpipelined CPU. ( )

b. Control hazard can be removed by forwarding in pipelined MIPS processor. ( )

c. Forwarding is used to reduce the pipeline stall due to read-after-write hazards. ( )

d. If you can predict the branch target perfectly, you can remove the control hazards completely. ( )

e. Set-associative cache reduces the capacity misses. ( )

f. Direct mapped cache offers better hit time than set-associative caches. ( )

g. Fully set-associativity cache does not use index bits during search. ( )

h. The virtual address space can be larger than the physical address space. ( )

i. Increasing cache block size can take advantage of temporal locality. ( )

j. Conflict misses can be completed removed by the fully-asssociative cache . ( )

2. Pipelined processor [25 points]

Please consider the following 7 stage pipelined processor (depicted below). The pipeline is similar to the standard 5 stage pipeline with the exception that the icache (IF stage) and dcache (MEM stage) are both pipelined with a latency of 2 cycles. Results from cache reads are available at the end of 2 cycles. As usual, the branch compare is calculated and the new PC loaded during the ID stage.

IF1	IF2	ID	EX	MEM1	MEM2	WB

Consider the following codes.

```
Loop:  lw $2, 0($4)
       add $1,$1,$2
       add $5,$5,$1
       addi $4,$4,4
       addi $3,$3,-1
       bgez $3, Loop
```

(a) **[5 points]** Identify all read-after-write dependencies in the code above. You should draw circles around two registers that exhibit the dependency and connect them with an arrow.

(b) [10 points] Assume the pipelined processor provides full forwarding and a hazard detection logic. Please **fill the multi-cycle pipelined diagram**. Stall cycles can be expressed by inserting the dash (--). Note that MEM1/2 are same as M1/M2.

Lw	IF1	IF2	ID	EX	M1	M2	WB											
Add																		
Add																		
Addi																		
Addi																		
Bgez																		

(c) [7 points] To reduce the stall cycle, the compiler can reschedule the code. Please write down the best optimized code sequence.

(d) [8 points] Clock cycle time is reduced from 1 ns to 0.6 ns by further pipelining 5 MIPS pipelines into 7 pipelines as in (b). Considering the code in (a), does this deeper pipelining provide better performance? Justify your answers by comparing total execution time for one iteration of the loop. Ignore the control hazards due to the branch instruction.

3. Code Evaluation for Cache [30 points]

Consider the following C code which performs a matrix transpose.

```
int a[16][16],b[16][16];
for (i=0;i<16;i++)
  for (j=0;j<16;j++)
      a[j][i] = b[i][j];
```

An array is stored in the memory in row major order as follows.

address	Data
0x1000	A[0][0]
0x1004	A[0][1]
0x1008	A[0][2]
…	…
0x1040	A[1][0]
0x1044	A[1][1]
…	…

Suppose that array a is located at address 0x1000 and array b is located at address 0x1400.The processor has a direct mapped data cache whose size is 4 KB and the size of the cache block is 16 bytes.

(a) [5 points] How many bits are allocated for the tag, index, and block offset of the cache assuming we have a 32-bit memory address (note that a block offset does not include a byte offset)?

b) [5points] How many load and store instruction per innermost loop iteration?

c) [10 points] How many data cache misses for the matrix ‘a’ during executing the above code when a direct mapped cache is used? Show your work. (No credit for just an answer.)

d) [5 points] Assume that matrix ‘a’ is now a 512 x 512 matrix (e.g. a[512][512]) and the loop index range of the code is adjusted accordingly. What is the miss rate for accessing the matrix ‘a’ during the execution of the code. Ignore the matrix ‘b’ for this analysis (assume that you do not have the matrix ‘b’). Justify your answers for the full credit.

e) [5 points] If you have a two-way set-associative cache for the data cache, does that improve the miss rate of the matrix ‘a’ in (d)? You need to explain your answer to the full credit.

4. Cache Performance [15 points]

Please consider the following processor and cache configurations.

Clock rate	500MHz
Base CPI	1.0
L1 cache type	Separate Instruction and Data Cache
L1 cache write policy	Write-back
L1 cache hit time	2 ns
L1 miss penalty	40ns (time to read from and write to main memory)
L1 instruction cache miss rate	1.0%
L1 data cache miss rate	5%
dirty data cache blocks	40 % of total data cache blocks

(a) [5 points] Compute the average memory access time (AMAT) for Instruction cache (in ns).

(b) [10 points] Compute the average memory access time (AMAT) for Data cache (in ns) assuming load and store represent 20 % of instructions.