CS 3410 Fall 2026

Lab 8 Worksheet

1. Cache of Clues: Tag, You're it!

Suppose we have a **16-byte** direct-mapped cache with **4-byte** blocks, and **8-bit** memory addresses

a. How many total Cache Blocks are there? → _____
b. How many offset bits are needed? → _____
c. How many index bits are needed? → _____
d. How many tag bits remain? → _____

2. Do the Math, Cache the Glory

A cache has four primary configuration parameters:

- **A**: Associativity (number of ways per set)
- **B**: Block Size (size of a single block, a block is also referred to as a cache line)
- **C**: Capacity or cache size (total amount of data in the cache)
- **W**: Number of bits in each memory address

These three parameters determine the number of sets and ways in the cache. These parameters also specify how a memory reference address is split into its three components: tag, index, and offset.

Write formulas in terms of **A**, **B**, **C, W** to calculate the following quantities. You may find using log(A), log(B), and log(C) easier than A, B, and C directly. You may assume A, B, and C are all powers of 2.

a. Number of offset bits in the address → _____
b. Number of sets in the cache → _____
c. Total number of cache lines in the cache → _____
d. Number of index bits in the address → _____
e. Number of tag bits in the address → _____

3. AMAT (Average Memory Access Time): Race Against Latency

a. Calculate the average memory access time for the following memory organization.

Memory Level	**Access Time (Cycles)**	**Miss Rate**
Cache	1	10%
Main Memory	100	0%

AMAT: _____ cycles

b. An L2 cache is added to the system. Calculate the new average memory access time.

Memory Level	Access Time (Cycles)	Miss Rate
L1 Cache	1	10%
L2 Cache	10	10%
Main Memory	100	0%

AMAT: _____ cycles

4. The Fast and the Cache-ious

You are given a direct-mapped cache with **4 lines**. Each cache line stores **1 byte of data**. The address is divided into tag|index, where the two LSB are used as the index. The main memory contains **16 bytes**, with addresses from 0000 to 1111. For each of the memory access, fill in the table.

Instruction	Index	Tag	Hit/Miss	Type of Miss
load 1100				
load 1101				
load 0100				
load 1100				

MEMORY

addr	data
1111	P
1110	O
1101	N
1100	M
1011	L
1010	K
1001	J
1000	I
0111	H
0110	G
0101	F
0100	E
0011	D
0010	C
0001	B
0000	A

5. Ready, Set, Associate!

You are given an 8-byte, 2-way set-associative cache. Assume block sizes are 2 bytes and total memory is 16 bytes. The cache starts empty. Use LRU (Least Recently Used) for replacement. For each of the memory access, fill in the table.

Instruction	Set (Index)	Tag	Offset	Hit/Miss	Eviction (Y/N)	Data Loaded
load 1011						
load 0110						
load 1010						
load 1110						
load 1010						