

CSC/ECE 506: Architecture of Parallel Computers
Problem Set 2
Due: Monday, June 23, 2025
Reflection Due: Monday, June 30, 2025

There will be two submission deadlines for this problem set. For the first submission deadline, you should turn in solutions to all the problems. Then the “official” solutions will be distributed. For the second submission deadline, you should turn in corrections of your work, and explanations of anything you got wrong. Your grade will depend mostly on the reflection.

There are 80 points on this problem set.

Problem 1. (20 points, 4, 4, 4, & 8 per part) For parts (a) to (c), assume the address width is 16 bits.

(a) If the block offset is 8 bits and 8 bits are tag bits, then what kind of cache mapping (direct mapped, set associative or fully associative) would it be?

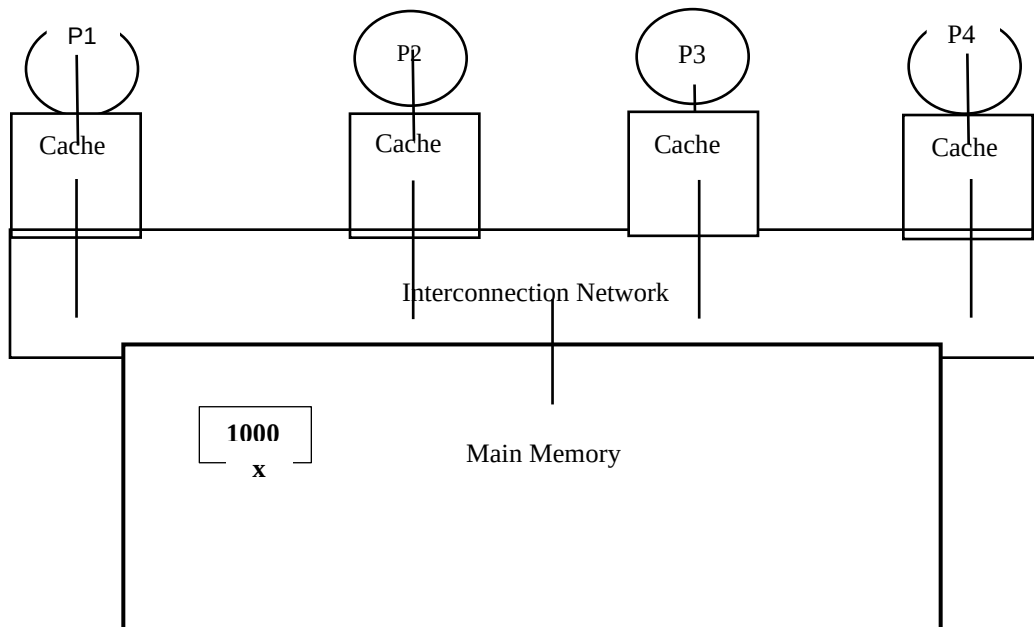
(b) If the cache size is 8 KB, and has an offset of 8 bits, with 5 index bits and the remaining bits being tag bits, then what kind of cache mapping (direct mapped, set associative or fully associative) would it be?

(c) Suppose the cache size is 8 KB, with an offset of 8 bits, 3 index bits and the remaining bits tag bits, then what kind of cache mapping (direct mapped, set associative or fully associative) would it be?

(d) To which set will a main-memory address of 9745_{16} be mapped, assuming that the cache contains 32 lines, with 16 bytes per line, assuming the following associativities?

- (i) direct mapped
- (ii) two-way set associative
- (iii) four-way set associative
- (iv) eight-way set associative?

Problem 2. (20 points) This problem relates to a system that has four processors (P1, P2, P3, and P4), each with a private L1 cache and a shared main memory. Memory location x is initialized to a decimal value of 1000. A logical arrangement of the system is shown below.



The system does not use a cache-coherence protocol.

The pseudo-assembly level code given below shows three threads run on processors P1, P2, P3 and P4 respectively.

Thread 1(P1)	Thread 2 (P2)	Thread 3 (P3)	Thread 4 (P4)
ld r2,x // r2=x ld r3,x // r3=x sub r1,r2,r3 // r1=r2-r3 st x, r1 // x=r1	ld r5, x // r5=x ld r4, x // r4 =x	 ld r6, x // r6=x	ld r7, x // r7=x <i>k: add r7,r7,100</i> <i>//r7=r7+100,</i> <i>k is the label.</i> ld r8, x // r8=x bnz r8,k <i>//branch to k,</i> <i>if r8 is non zero</i>
(a) Fill in the following table for the above sequence of operations, assuming write-back caches are used.			

Value of 'x' at different locations						Value of registers							
Action	Cache				Memory	P1			P2		P3	P4	
	P1	P2	P3	P4		r1	r2	r3	r4	r5	r6	r7	r8
Initially	-	-	-	-	1000	-	-	-	-	-	-	-	-
ld r2,x													
ld r3,x													
ld r5, x													
ld r7, x													
sub r1,r2,r3													
st x, r1													
add r7,r7,100													
ld r4, x													
ld r6, x													
ld r8,x													
Control Flow Instruction					Possible Outcome						Yes/No		
bnz r8,k					Will it branch to instruction 'k'								

- (b) Fill in the following table for the above sequence of operations, assuming write-through caches were used.

Value of 'x' at different locations						Value of registers							
Action	Cache				Memory	P1			P2		P3	P4	
	P1	P2	P3	P4		r1	r2	r3	r4	r5	r6	r7	r8
Initially	-	-	-	-	1000	-	-	-	-	-	-	-	-
ld r2,x													
ld r3,x													
ld r5, x													
ld r7, x													
sub r1,r2,r3													
st x, r1													
add r7,r7,100													
ld r4, x													
ld r6, x													
ld r8,x													
Control Flow Instruction					Possible Outcome						Yes/No		
bnz r8,k					Will it branch to instruction 'k'								

- (c) Does a write-back cache provide a coherent view of the memory location x? Please explain your answer.
- (d) Does using write through cache provide a coherent view of the memory location x? What improvement does it provide, when compared to write back caches? Please explain your answer.

Problem 3. (20 points) Here is a table of actions when Processors 1, 2, and 3 read or write the same cache block (held in each of the caches).

- (a) If an MSI protocol (without bus upgrade) is in use, what would be the states or actions in cells containing a ?.

Proc Action	State P1	State P2	State P3	Bus Action	Data From
W1	M	-	-	BusRdX	Mem
R1	M	-	-	-	Own Cache
W2	?	M	-	BusRdX	Mem
R1	S	S	-	?	?
W3	I	I	M	?	Mem
R1	S	I	?	BusRd	?
R3	S	I	S	-	Own Cache
R2	S	S	S	BusRd	?
?	M	I	I	BusRdX	Mem

(b) Suppose that MSI with bus upgrade is in use. How would the solution change?

Problem 4. (20 points) Below you will find two code segments. Determine what results are possible and what results are not, under sequential consistency. If a result is possible, list the execution sequence that will produce it. If a result is not possible, prove it is impossible. Assume all variables are initialized to 0.

(a) Processor 1	Processor 2
1a $x = 1$	2a $z = x$
1b $y = x$	2b $x = y$

(b) Processor 1	Processor 2
1a $x = y$	2a $x = 1$
1b $z = 2$	2b $y = x$