CO 2021-Fall HW5 Solution

5.8

5.8.1

Assuming the addresses given as byte addresses, each group of 32 accesses will map to the same 32-byte block so the cache will have a miss rate of 1/32. All misses are compulsory misses. The miss rate is not sensitive to the size of the cache or the size of the working set. It is, however, sensitive to the access pattern and block size.

5.8.2

The miss rates are 1/16, 1/64, and 1/128, respectively. The workload is exploiting spatial locality.

5.8.3

In this case the miss rate is 0.

5.9

5.9.1

AMAT for B = 8:	0.04 x (20 x 8) = 6.4
AMAT for B = 16:	0.03 x (20 x 16) = 9.6
AMAT for B = 32:	0.02 x (20 x 32) = 12.8
AMAT for B = 64:	0.01 x (20 x 64) = 19.2
AMAT for B = 128:	0.01 x (20 x 128) = 25.6

B = 8 is optimal.

5.9.2

AMAT for B = 8:	0.04 x (24 + 8) = 1.28
AMAT for B = 16:	0.03 x (24 + 16) = 1.2
AMAT for B = 32:	0.02 x (24 + 32) = 1.12
AMAT for B = 64:	0.01 x (24 + 64) = 1.32
AMAT for B = 128:	0.01 x (24 + 128) = 1.52

B = 32 is optimal.

5.9.3

B = 128

5.10

5.10.1

Clock rate P1: 1 / 0.66 ns = 1.515 GHz

Clock rate P2: 1 / 0.9 ns = 1.11 GHz

5.10.2

AMAT P1: 0.66 + 70 * 0.08 = 6.26 ns or 6.26 / 0.66 = 9.48 cycles

AMAT P2: 0.9 + 70 * 0.06 = 5.1 ns or 5.1 / 0.9 = 5.666 cycles

5.10.3

CPI P1 = 1 + 1.36 * 0.08 * 70 / 0.66 = 12.539

CPI P2 = 1 + 1.36 * 0.06 * 70 / 0.9 = 7.346

5.10.4

L2 global miss rate = 0.08 * 0.95 = 0.076

AMAT P1 = 0.66 + 0.08 * 5.62 + 0.076 * 70 = 6.4296

Worse

5.10.5

CPI P1 = 1 + 1.36 * (0.08 * 5.62 / 0.66 + 70 * 0.076 / 0.66) = 12.888

5.19

5.19.1

It would be invalid if it was paged out to disk.

5.19.2

A write to page 30 would generate a TLB miss. Software-managed TLBs are faster in cases where the software can pre-fetch TLB entries.

5.19.3

When an instruction writes to VA page 200, and interrupt would be generated because the page is marked as read only.

5.24

5.24.1

The cache should be able to satisfy the request since it is otherwise idle when the write buffer is writing back to memory. If the cache is not able to satisfy hits while writing back from the write buffer, the cache will perform little or no better than the cache without the write buffer, since requests will still be serialized behind writebacks.

5.24.2

Unfortunately, the cache will have to wait until the writeback is complete since the memory channel is occupied. Once the memory channel is free, the cache is able to issue the read request to satisfy the miss.

5.24.3

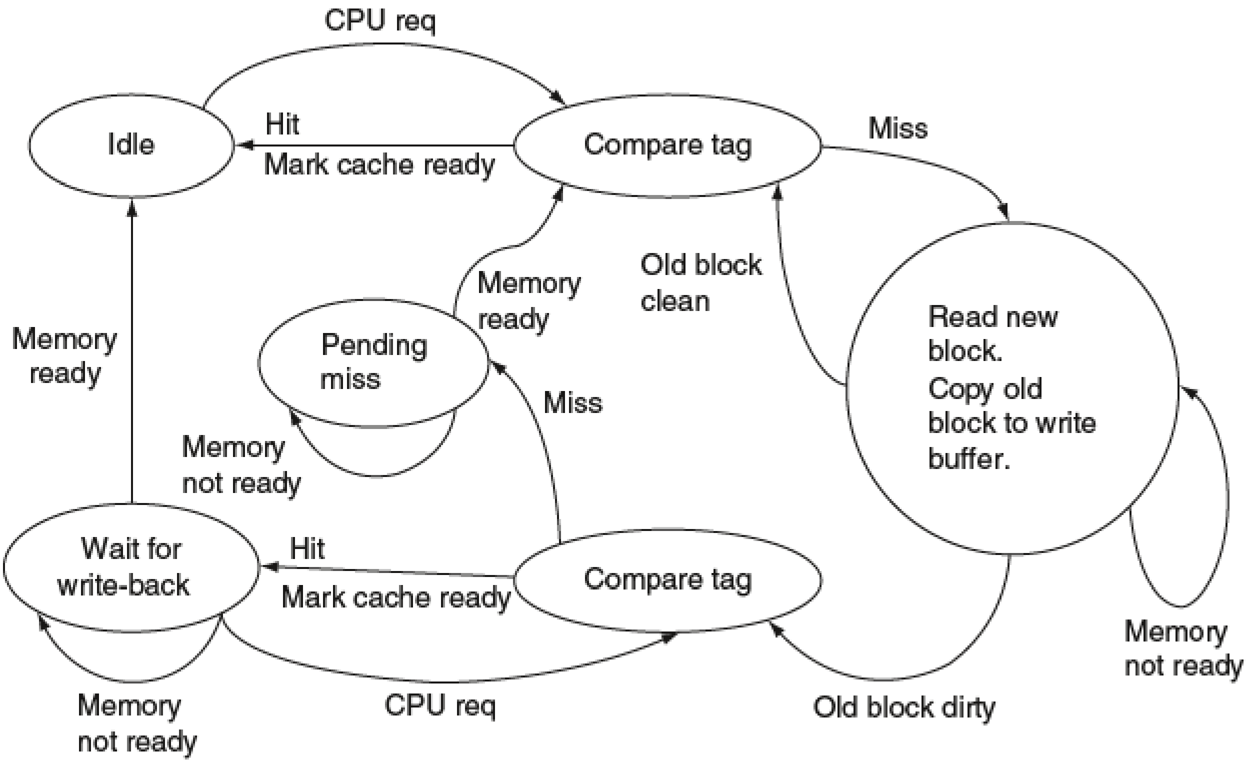
Correct solutions should exhibit the following features:

1. The memory read should come before memory writes.
2. The cache should signal “Ready” to the processor before completing the write.

Example (simpler solutions exist; the state machine is somewhat):

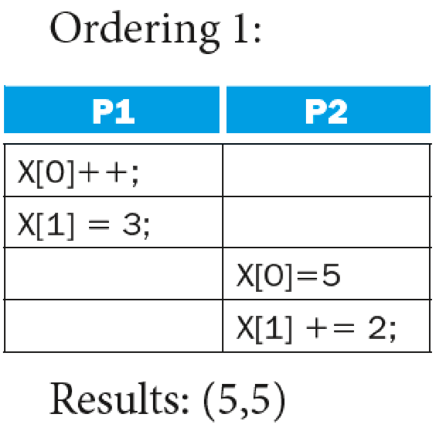
underspecified in the chapter):

5.24.3 cont.



5.25

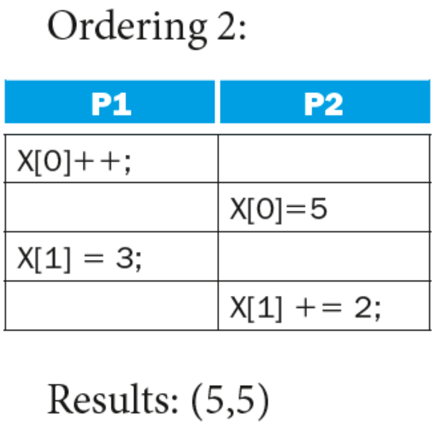
5.25.1



Ordering 1:

P1	P2
X[0]++;	
X[1] = 3;	
	X[0]=5
	X[1] += 2;

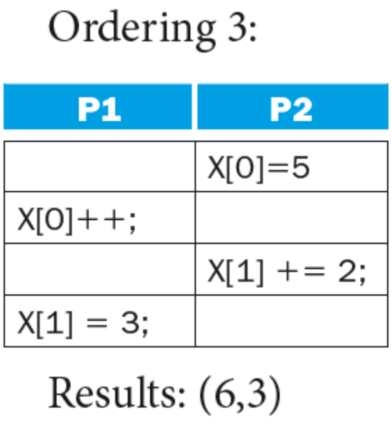
Results: (5,5)



Ordering 2:

P1	P2
X[0]++;	
	X[0]=5
X[1] = 3;	
	X[1] += 2;

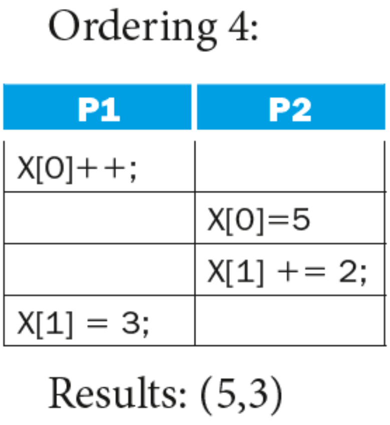
Results: (5,5)



Ordering 3:

P1	P2
	X[0]=5
X[0]++;	
	X[1] += 2;
X[1] = 3;	

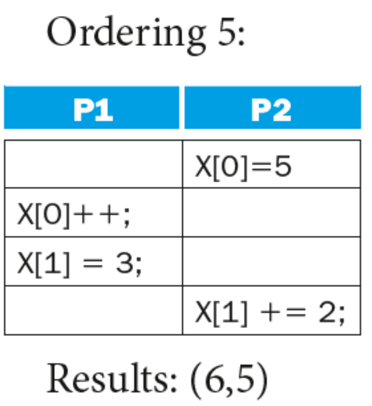
Results: (6,3)



Ordering 4:

P1	P2
X[0]++;	
	X[0]=5
	X[1] += 2;
X[1] = 3;	

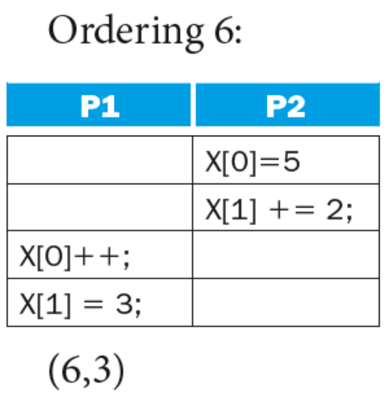
Results: (5,3)



Ordering 5:

P1	P2
	X[0]=5
X[0]++;	
X[1] = 3;	
	X[1] += 2;

Results: (6,5)



Ordering 6:

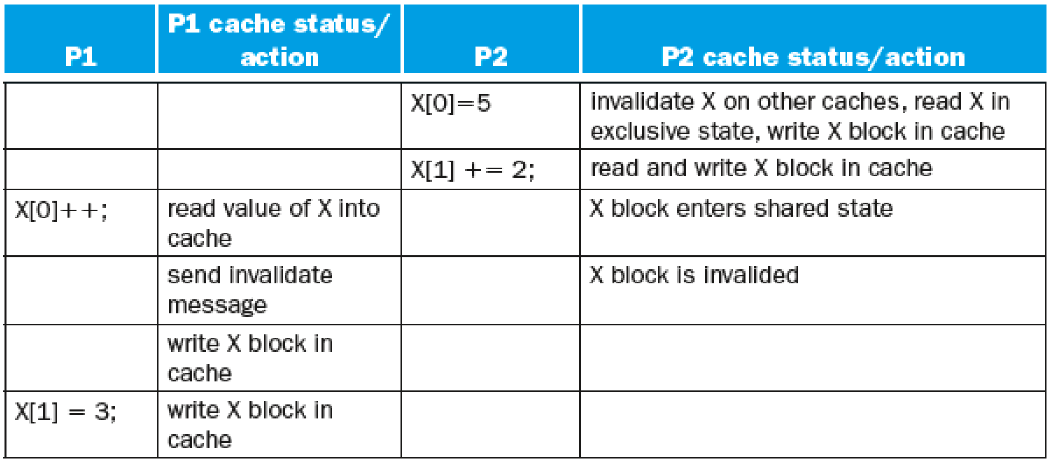
P1	P2
	X[0]=5
	X[1] += 2;
X[0]++;	
X[1] = 3;	

(6,3)

If coherency isn't ensured:

P2's operations take precedence over P1's: (5,2)

5.25.2



P1	P1 cache status/ action	P2	P2 cache status/action
		X[0]=5	invalidate X on other caches, read X in exclusive state, write X block in cache
		X[1] += 2;	read and write X block in cache
X[0]++;	read value of X into cache		X block enters shared state
	send invalidate message		X block is invalided
	write X block in cache		
X[1] = 3;	write X block in cache		

5.25.3

Best case:

Orderings 1 and 6 above, which require only two total misses.

Worst case:

Orderings 2 and 3 above, which require 4 total cache misses.