Introduction to Computer Architecture

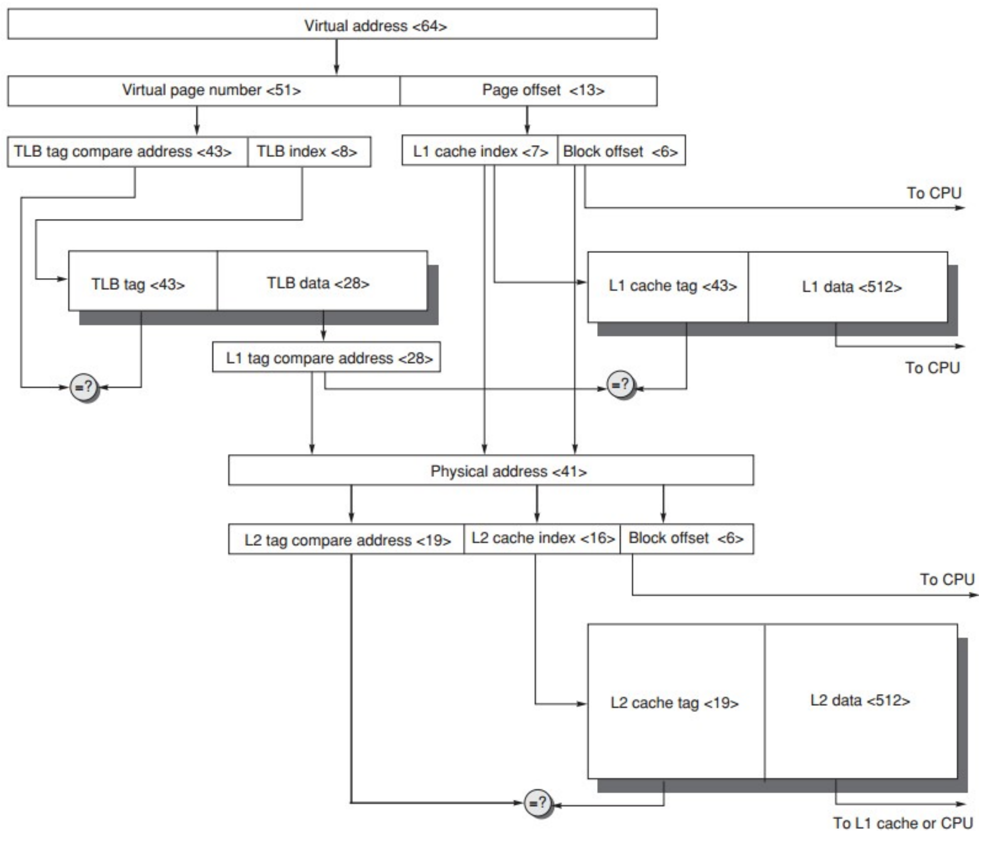
Assignment 2

Due January 03, 2018

1. [10 = 8 + 2]
Assume the following RAID 3 system uses even parity and disk2 fails. Determine the original data on disk 2 and fill the form.
Use the first two rows as example to explain how the original data is recovered.

Disk 1	Disk 2 - Failed	Disk 3	Parity
1		1	1
0		0	1
1		1	0
0		0	0
0		0	1
0		0	1
1		1	0
1		1	1

2. [10]
Describe the address translation process on the following memory system.



3. [10]
Assume a fully associative write-back cache with many cache entries that starts empty. Below is a sequence of five memory operations (the address is in square brackets):

Write	Mem[100];
Write	Mem[100];
Read	Mem[200];
Write	Mem[200];
Write	Mem[100];

What are the number of hits and misses (and which are them) when using no-write allocate versus write allocate?

4. [10]
For the code below, assume we have an 8 KB direct-mapped data cache with 16-byte blocks, and it is a write-back cache that does write allocate. The elements of a and b are 8 bytes long since they a double-precision floating-point arrays. There are 3 rows and 100 columns for a and 101 rows and 3 columns for b. Assume they are not in the cache at the start of the program.
Determine the number of cache misses and which accesses cause them for the following codes with or without prefetching.

```
for (i = 0; i < 3; i = i+1)
        for (j = 0; j < 100; j = j+1)
                a[i][j] = b[j][0] * b[j+1][0];
```

```
for (j = 0; j < 100; j = j+1) {
        prefetch(b[j+7][0]);
        /* b(j,0) for 7 iterations later */
        prefetch(a[0][j+7]);
        /* a(0,j) for 7 iterations later */
        a[0][j] = b[j][0] * b[j+1][0];};
for (i = 1; i < 3; i = i+1)
        for (j = 0; j < 100; j = j+1) {
                prefetch(a[i][j+7]);
                /* a(i,j) for +7 iterations */
                a[i][j] = b[j][0] * b[j+1][0];}
```

5. [10]
Look at this code sequence:

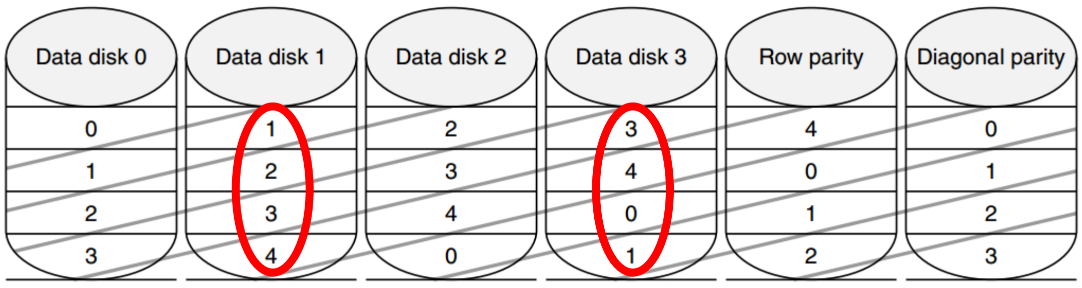
SW	R3,	512(R0);	M[512] <- R3	(cache index 0)
LW	R1,	2014(R0);	R1 <- M[1024]	(cache index 0)
LW	R2,	512(R0);	R2 <- M[512]	(cache index 0)

Assume a direct-mapped write-through cache that maps 521 and 1024 to the same block, and a four-word write buffer that is not checked on a read miss. Will the value in R2 always be equal to the value in R3? Explain why.

6. [10 = 5 + 5]
Suppose that in 1000 memory references there are 40 misses in the first-level cache and 20 misses in the second-level cache. Assume the miss penalty from the L2 cache to memory is 200 clock cycles, the hit time of the L2 cache is 10 clock cycles, the hit

time of L1 is 1 clock cycle, and there are 1.5 memory references per instruction.
***a**. What is the average memory access time and*
***b.** average stall cycles per instruction?*
Ignore the impact of writes.

7. [10]
Provide a suitable recovery sequence for the illustrated RAID-DP (or row-diagonal parity) with disks 1 and 3 failed.



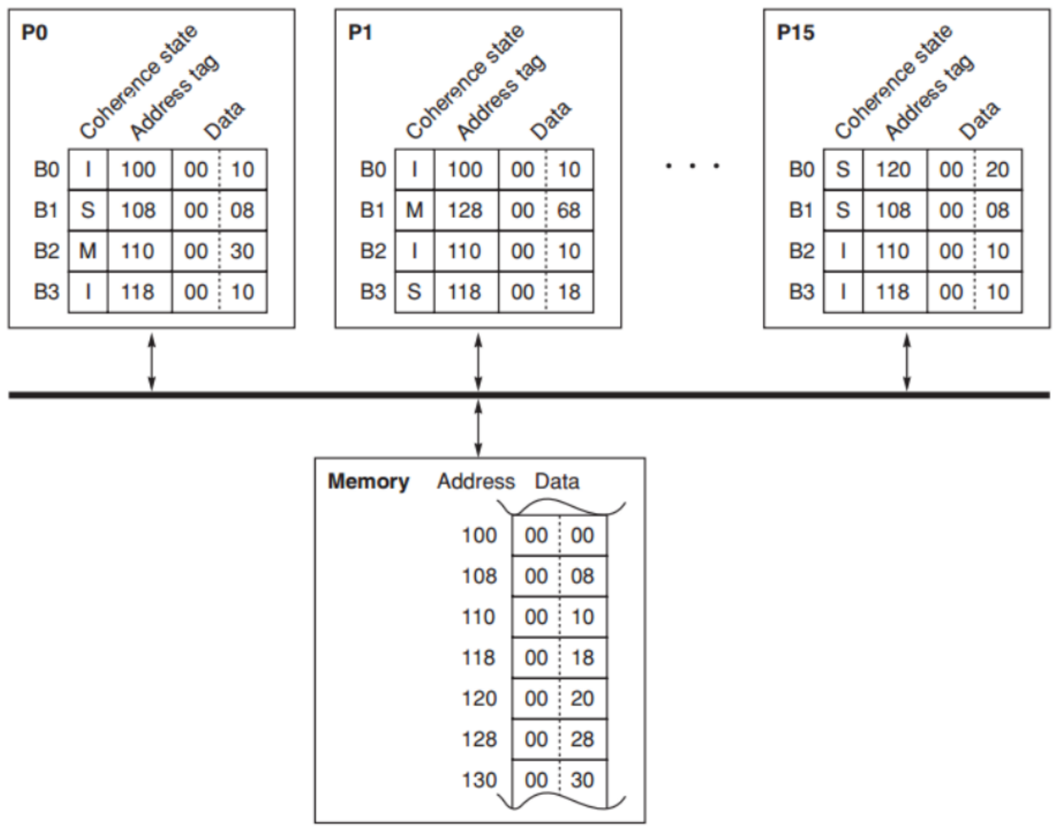
8. [10]
To be covered on January 03, please complete this question during class.

Assume that words x1 and x2 are in the same cache block, which is in the shared state in the caches of P1 and P2. Assuming the following sequence of events, identify each miss as a true sharing miss, a false sharing missing, or a hit. Any miss that would occur if the clock size were one word is designated a true sharing miss.

Time	P1	P2
1	Write x1	
2		Read x2
3	Write x1	
4		Write x2
5	Read x2	

9. [20 = 5 x 4]
To be covered on January 03, please complete this question during class.

The simple, bus-based multiprocessor illustrated in the following figure represents a commonly implemented symmetric shared-memory architecture. Each processor has a single, private cache with coherence maintained using the snooping coherence protocol. Each cache is directed-mapped, with four blocks each holding two words. To simplify the illustration, the cache-address tag contains the full address and each word shows only two hex characters, with the least significant word on the right. The coherence states are denoted M, S, and I for Modified, Shared, and Invalid.



Treat each action below as independently applied to the initial state as given in the figure.
What is the resulting state (i.e., coherence state, tags, and data) of the caches and memory after the given action? Show only the blocks that change, for example, P0.B0: (I, 120, 00 01) indicates that CPU P0's block B0 has the final state of I, tag of 120, and data words 00 and 01.
Also, what value is returned by each read operation?

a. P0: read 120
b. P0: write 120 <-- 80
c. P15: write 120 <-- 80
d. P1: read 110