Introduction to Computer Architecture

Assignment 3

Due January 07, 2019

Merry Christmas & Happy New Year

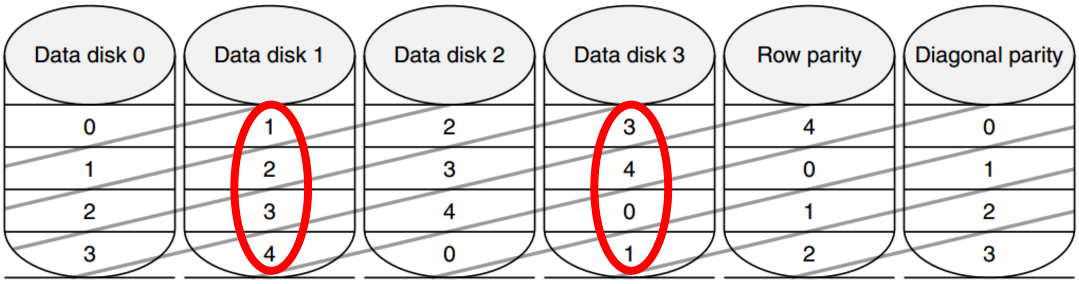
1. [5 = 3 + 2]

Assume that the following RAID 3 system uses odd parity and disk2 fails.
a. Determine the original data on disk 2 and fill the form.
b. Use the first two rows as example to explain how the original data are 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. [5]

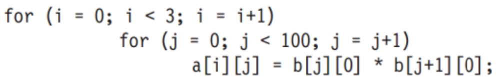
Provide a suitable recovery sequence for the illustrated RAID-DP (or row-diagonal parity) with disks 1 and 3 failed.
Provide also the specific derivation process.



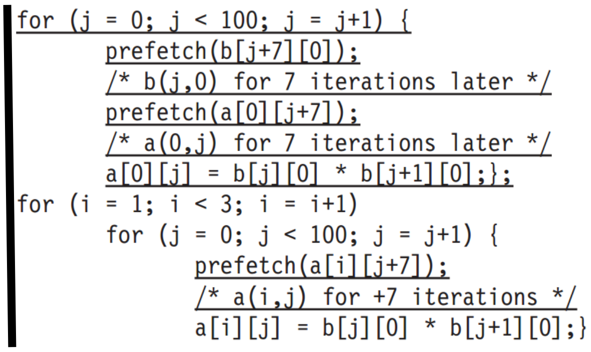
3. [10]

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];}
```

4. [5]

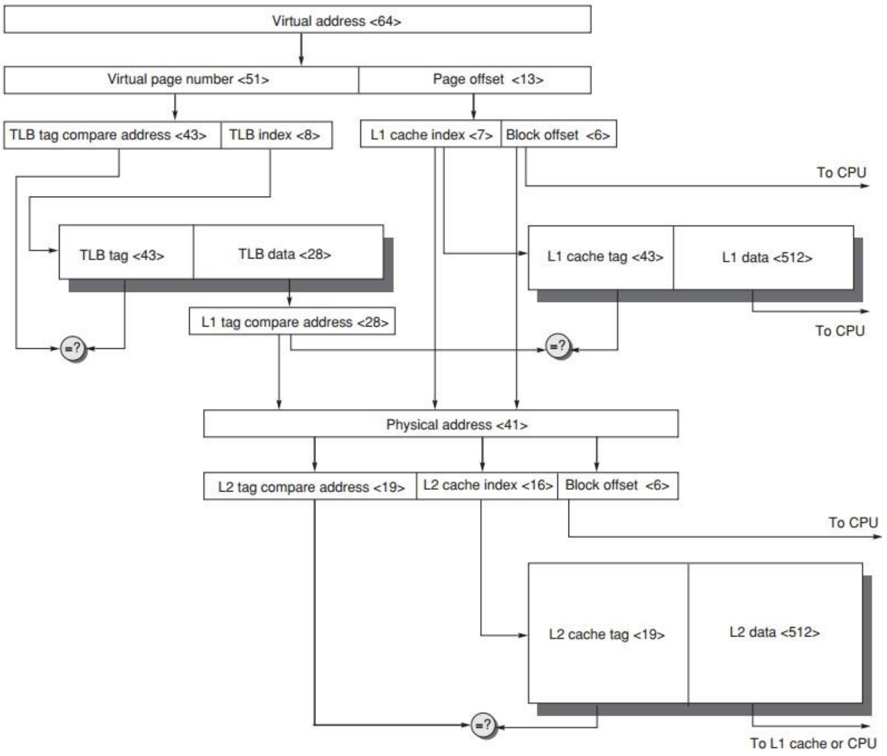
Look at this code sequence:

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.

5. [5]

Describe the address translation process on the following memory system.



6. [5]

Assume:
A processor has a direct mapped cache;
Data words are 8 bits long;
Data addresses are to the word;
A physical address is 20 bits long;
The tag is 11 bits;
Each block holds 16 bytes of data.

How many blocks are in this cache?

7. [10]

Consider a 16-way set associative cache:
Data words are 64 bits long;
Words are addressed to the half-word;
The cache holds a 2 Mbytes of data;
Each block holds 16 data words;
Physical addresses are 64 bits long.

How many bits of tag, index, and offset are needed to support references to this cache?

8. [10]

The average memory access time for a microprocessor with 1 level of cache is 2.4 clock cycles. If data are present and valid in the cache, it can be found in 1 clock cycle. If data are not found in the cache, 80 clock cycles are needed to get it from off-chip memory.

Designers are trying to improve the average memory access time to obtain a 65% improvement in average memory access time, and are considering adding a second level of cache on-chip. This second level of cache could be accessed in 6 clock cycles. The addition of this cache does not affect the first level cache's access patterns or hit times. Off-chip accesses would still require 80 additional clock cycles.

To obtain the desired speedup, how often must data be found in the second level cache?

9. [10]

When parallelizing an application, the ideal speedup is speeding up by the number of processors. This is limited by two things: percentage of the application that can be parallelized and the cost of communication. Amdahl’s law takes into account the former but not the later.

a. What is the speedup with N processors if 80% of the application is parallelizable, ignoring the cost of communication?

b. What is the speedup with 8 processors if, for every processor added, the communication overhead is 0.5% of the original execution time?
c. What is the speedup with 8 processors if, for every time the number of processors is doubled, the communication overhead is increased by 0.5% of the original execution time?
d. What is the speedup with N processors if, for every time the number of processors is doubled, the communication overhead is increased by 0.5% of the original execution time?
e. Write the general equation that solves this question: What is the number of processors with the highest speedup in an application in which P% of the original execution time is parallelizable, and, for every time the number of processors is doubled, the communication is increased by 0.5% of the original execution time?

10. [5]

Show how the instructions in the sequence given below will proceed through the pipeline
:: We will predict that the beq instruction is not taken
:: When the beq instruction is executed, the value in $1 is equal to the value in $2

clock cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
beq $1, $2, **X**															
lw $10, 0($11)															
sub $14, $10, $10															
X: add $4, $1, $2															
lw $1, 0($4)															
sub $1, $1, $1															
add $1, $1, $1															

11. [10]

Assume a 4 Gbytes main memory. 1 Gbyte of the 4 Gbytes has been reserved for process page table storage.
Each page table entry consists of:
:: A physical frame number;
:: 1 valid bit;
:: 1 dirty bit;
:: 1 LRU status bit;
Virtual addresses are 32 bits. Physical addresses are 26 bits. The page size is 8 Kbytes.

How many process page tables can fit in the 1 Gbyte space? Assume that each page table entry contains three extra bits: a valid bit, a dirty bit, and an LRU bit.

12. [5]

Which topic do you think should be covered with more details in the review class? Why? What's your current understanding of that topic? What's the unclear or ambiguous part?

13. [5]

Given the following sequence of pipelined instructions:

```
lw      $1, 0($2)
add     $3, $4, $5
sub     $6, $1, $3
```

And the following diagram of pipeline datapath:

IF	IF/ID latch	ID	ID/EX latch	EX	EX/Mem latch	Mem	Mem/WB latch	WB

Where will the data operands that are processed during the EX stage of the subtract (sub) instruction come from?

666. [10] It's Your Turn

Design a question that you think is feasible as an exam question.

a. which topic you would like to consider, pipeline, cache, memory, or storage?

b. describe the (sufficiently complex) question; (If you are familiar with pipeline datapath, you are encouraged to design related questions.)

c. provide also a *correct* sample solution, thanks.

Sample solution from one dalao in the previous class:

a. Class

b. Who gives the class candies?

c. Solution: X W T

Warning: You can ask X W T for candies. You are, however, not expected to provide similar solutions.

THANK YOU & SEE YOU IN 9102