In the name of God

HOMEWORK #5

Computer Architecture

Dr.Beitolahi

Spring 1401

Designer : Marziye Pandi, Mostafa Moghaddas

Deadline :
1401/3/30

Notes :

1. Do the homeworks with your own information.

2. Any similarity between solutions reduces 1 point from overall point of your assignments.

3. No submissions will be accepted after the announced deadline.

1- Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of 32-bit memory address references, given as word addresses.

3, 180, 43, 2, 191, 88, 180, 14, 181, 44, 186, 253

a- For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. Also lost if each reference is a hit or a miss, assuming the cache is initially empty.

b- For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with two-word blocks and a total size of 8 blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

c- You are asked to optimize a cache design for the given references. There are three direct-mapped cache designs possible, all with a total of 8 words of data: C1 has 1-word blocks, C2 has 2-word blocks, and C3 has 4-word blocks. In terms of miss rate, which cache design is the best? If the miss stall time is 25 cycles, and C1 has an access time of 2 cycles, C2 takes 3 cycles, and C3 takes 5 cycles, which is the best cache design?

2- Virtual memory uses a page table a track the mapping of virtual addresses to physical addresses. This exercise shows how this table must be updated as addresses are accessed. The following data constitutes a stream of virtual addresses as seen on system. Assume 4KB pages, a 4-entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number.

4669, 2227, 13916, 34587, 48870, 12608, 49225

TLB

Valid	Tag	Physical Page Number
1	11	12
1	7	4
1	3	6
0	4	9

Page table

Valid	Physical Page or in Disk
1	5
0	Disk
0	Disk
1	6
1	9
1	11
0	Disk
1	4
0	Disk
0	Disk
1	12
1	3

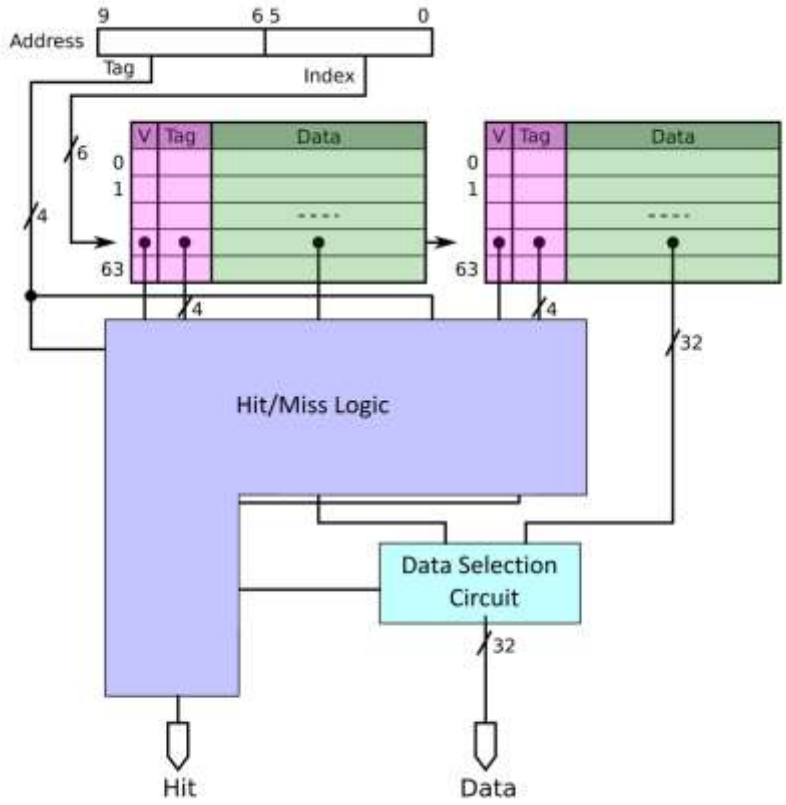
a- Given the address stream shown, and the initial TLB and page table states provided above, show the final state of the system. Also list for each reference if it is a hit in the TLB, a hit in the page table, or a page fault.

b- Repeat a, but this time use 16KB pages instead of a 4KB pages. What would be some of the advantages of having a larger page size? What are some of the disadvantages?

c- Show the final contents of the TLB if it is 2-way set associative. Also show the contents of the TLB if it is direct mapped. Discuss the important of having a TLB to high performance. How would virtual memory accesses be handled if there were no TLB?

Implementation:

A 2-way set-associative cache is shown in the following figure. Each set of this memory has 64 lines, and each line contains a single 4-byte word. The cache rows are selected based on the index. Following the selection of the desired row, the tag part of the input data is compared with the tag part of the data in the corresponding lines of the cache. The cache declares a hit when the entries match, otherwise, it announces a miss. In the event of a hit, the correct data is selected by a circuit and is given to the output.



This cache is implemented as a combination of several parts that are implemented individually.

It consists of a data array, a tag-valid array, a miss-hit logic array, a LRU array, and a cache controller.

As a matter of convenience, in this exercise you only have to implement the Miss-Hit logic section.

Miss-Hit Logic: The purpose of this module is to determine whether a hit has occurred. When a hit occurs, this module is also tasked with identifying in which sets it occurred. This module is illustrated in the following figure.

It should be noted that w0 and w1 represent the tag and valid bits of the first set and the second set, respectively. The tag signal is the same tag we use as input to the cache (i.e., the same tag that we search for in the cache).

Let us say we wish to check whether data with a tag of 4 and an index of 20 exist in the cache, i.e., Tag = 0100. Therefore, we refer to the twentieth cache line.

Imagine that in the twentieth line of the cache, there is data with tag 7 and valid = 1 in way-0 and data with tag 4 and valid = 1 in way-1. As a result, the signal w0 equates to 10111 and that of w1 equates to 10100. Due to the fact that the data in way-1 is what we are trying to locate, we will have w0_valid = 0, w1_valid = 1, and hit = 1. The blue color indicates the valid, while the red color indicates the tag. If data is contained in the first set, the signal W0_valid is activated. This also applies for the signal W1_valid.