Assignment 3

CIS*2030: Structure and Application of Microcomputers

Assignment 3 is due **11:59pm, December 3, 2021**. The assignment is to be completed individually. If you have any questions, please see the teaching assistant or instructor during office hours.

Instructions:

(1) When answering questions, simply writing down a solution to a particular problem is not enough to obtain full marks. You must show how you arrived at the solution by showing the step-by step procedure that you performed. This procedure can be hand-written or typed – the choice is yours. However, solutions must be clear. What cannot be read cannot be marked.
(2) Make sure to neatly print/type your full name, student ID, and the name of your teaching assistant at the top of the first page, number the remaining pages consecutively at the bottom.
(3) Review your answers 1 or 2 days after completing the assignment. Taking a few minutes to do this can dramatically improve your understanding of key concepts and increase your retention. Also, it makes it much easier for you to catch any errors.

Submission:

(1) Once you are ready to submit your assignment for grading print or scan your individual assignment pages and convert them into a *single* PDF submission. (Do not upload individual pages or a zip file.) If you are working with paper and do not have a scanner, use a cellphone to take pictures of each page, then use a program like Adobe Acrobat Scan to create a single PDF. (Remember to use appropriate lighting, as dim or blurry pictures that cannot be read cannot be marked.) Upload your single PDF submission to the Dropbox labeled **Assignment 3** available in the **Topic 12** folder on CourseLink.
(2) After uploading your assignment to CourseLink, immediately download the assignment and verify that you have uploaded the correct document.

Best of success! 😊

Questions

This assignment covers general concepts related to **caching**, as discussed in *weeks 11* and *12*.

1. In the context of a running program, spatial locality arises due to the sequential execution of instructions. Is the previous statement true or false. Explain. [1 point]

2. In the context of a running program, temporal locality arises due to the execution of loops. Is the previous statement true or false. Explain. [1 point]

3. A particular cache memory has the following parameters: (1) a size of 1024 bytes not including overhead such as the valid and tag bits, (2) a block size of 4 bytes, and (3) 1 line per set. If the number of physical main memory address bits is 32, determine the following for the cache:

   a) The number of cache sets. [1/2 point]
   b) The number of index bits. [1/2 point]
   c) The number of block offset bits. [1/2 point]
   d) The number of tag bits. [1/2 point]

4. A particular cache memory has the following parameters: (1) a size of 1024 bytes not including overhead such as the valid and tag bits, (2) a block size of 8 bytes, and (3) 4 lines per set. If the number of physical main memory address bits is 32, determine the following for the cache:

   a) The number of cache sets. [1/2 point]
   b) The number of index bits. [1/2 point]
   c) The number of block offset bits. [1/2 point]
   d) The number of tag bits. [1/2 point]

5. A particular cache memory has the following parameters: (1) a size of 1024 bytes not including overhead such as the valid and tag bits, (2) a block size of 32 bytes, and (3) 32 lines per set. If the number of physical main memory address bits is 32, determine the following for the cache:

   a) The number of cache sets. [1/2 point]
   b) The number of index bits. [1/2 point]
   c) The number of block offset bits. [1/2 point]
   d) The number of tag bits. [1/2 point]

The problems that follow assume the following: (1) memory is byte addressable, (2) memory references are to 1-byte words (not to 4-byte words), (3) addresses are 12 bits wide, (4) the cache is 2-way set associative, with a 4-byte block size and eight sets, and (5) the contents of the cache are as follows:

2-way set associative cache

	Line 0						Line 1					
Set index	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3
0	09	1	86	30	3F	10	00	0	—	—	—	—
1	45	1	60	4F	E0	23	38	1	00	BC	0B	37
2	EB	0	—	—	—	—	0B	0	—	—	—	—
3	06	0	—	—	—	—	32	1	12	08	7B	AD
4	C7	1	06	78	07	C5	05	1	40	67	C2	3B
5	71	1	0B	DE	18	4B	6E	0	—	—	—	—
6	91	1	A0	B7	26	2D	F0	0	—	—	—	—
7	46	0	—	—	—	—	DE	1	12	C0	88	37

All numbers are in hexadecimal

6. As stated above, memory addresses are 13 bits; that is, B_{15}-B_0. Using the previous notation, indicate what bits would be used to identify:

   a) The cache block offset. [1/2 point]
   b) The cache set index. [1/2 point]
   c) The cache tag. [1/2 point]

7. Now assume that a running program references the 1-byte word at address 0x0E34. Indicate the following: [1 mark each]

   a) Cache block offset: 0x_____ [1/2 point]
   b) Cache set index: 0x_____ [1/2 point]
   c) Cache tag: 0x_____ [1/2 point]
   d) Cache hit? (Y/N) _____ [1/2 point]
   e) Cache byte returned 0x_____ [1/2 point]

Note: Indicate the cache entry accessed and the cache byte value returned in hexadecimal. Indicate whether a cache miss occurs. If there is a cache miss, enter "M" for "Cache byte returned."

8. Now assume that a running program references the 1-byte word at address 0x0DD5. Indicate the following:

   a) Cache block offset: 0x_____ [1/2 point]
   b) Cache set index: 0x_____ [1/2 point]
   c) Cache tag: 0x_____ [1/2 point]

d) Cache hit? (Y/N) _____ [1/2 point]
e) Cache byte returned 0x_____ [1/2 point]

9. Now assume that a running program references the 1-byte word at address 0x1FE4. Indicate the following:

a) Cache block offset: 0x_____ [1/2 point]
b) Cache set index: 0x_____ [1/2 point]
c) Cache tag: 0x_____ [1/2 point]
d) Cache hit? (Y/N) _____ [1/2 point]
e) Cache byte returned 0x_____ [1/2 point]

10. For the previous cache, list all of the hex memory addresses that will hit in set 3. [1 point each]

11. For a particular system, a read request takes 3 nanoseconds on a cache hit, but 29 nanoseconds on a cache miss. Now assume that while a programming is executing, 88 percent of the processor's read requests result in a cache hit. What is the average read access time in nanoseconds? [2 points]