CS224 - Spring 2022 - Lab #6 (Version 2: April 19, 22:49)

Examining the Effect of Cache Parameters and Program Factors on Cache Hit Rate

Dates:

Section 1: Mon, 25 Apr, 8:30-12:20 in EA-Z04
Section 2: Wed, 27 Apr, 13:30-17:20 in EA-Z04
Section 3: Tue, 26 Apr, 13:30-17:20 in EA-Z04
Section 4: Fri, 29 Apr, Fri 08:30-12:20 in EA-Z04
Section 5: Wed, 27 Apr, 8:30-12:20 in EA-Z04
Section 6: Fri, 29 Apr, 13:30-17:20 in EA-Z04

TAs;Tutor:

Section 1: Pouya Ghahramanian, Pouria Hasani; Fazıl Keskin
Section 2: Alper Şahıstan, Hüseyin Eren Çalık; Burak Öçalan
Section 3: Kemal Büyükkaya, Kenan Çağrı Hırlak
Section 4: Pouria Hasani, Sepehr Bakhshi
Section 5: Kenan Çağrı Hırlak, Soheil Abadifard; Alper Mumcular
Section 6: Alper Şahıstan, Soheil Abadifard

TA name (x No of labs): email address
Alper Şahıstan (x2): alper.sahistan@bilkent.edu.tr
Hüseyin Eren Çalık (x1): eren.calik@bilkent.edu.tr
Kemal Büyükkaya (x1): kemal.buyukkaya@bilkent.edu.tr
Kenan Çağrı Hırlak (x2): cagri.hirlak@bilkent.edu.tr
Pouria Hasani (x2): pouria.hasani@bilkent.edu.tr
Pouya Ghahramanian (x1): ghahramanian@bilkent.edu.tr
Sepehr Bakhshi (x1): sepehr.bakhshi@bilkent.edu.tr
Soheil Abadifard (x2): soheil.abadifard@bilkent.edu.tr

Tutor name (x No of labs): email address
Alper Mumcular (x1): alper.mumcular@ug.bilkent.edu.tr
Burak Öçalan (x1): burak.ocalan@ug.bilkent.edu.tr (Tutoring in the lab Thursdays, 17:40 - 19:25)
Fazıl Keskin (x1): fazil.keskin@ug.bilkent.edu.tr

Purpose: Studying the effect of various cache design parameters. The first part includes problem solving and writing a program. The second part, the lab part, involves execution of the program (with possible extensions etc. if suggested by your TA) and preparing a report.

In your solutions and report make sure that you have proper tables, page numbering and understandable explanation with good writing. All tables must have a subtitle and table number. In tables columns must have meaning names/headers. Try to do everything before coming to the lab but make sure that you demonstrate your work to your TA.

Preliminary Work: 50 points

Preliminary Report/Preliminary Design Report: Involves problem solving related to cache memory design and a program written for cache testing.

Lab Work: 50 points

Experiments with caching and experiment report.

Important Notes for All Labs About Attendance, Performing and Presenting the Work

1. You are obliged to read this document word by word and are responsible for the mistakes you make by not following the rules.
2. Not attending to the lab means 0 out of 100 for that lab. If you attend the lab but do not submit the preliminary part you will lose only the points for the preliminary part.
3. Try to complete the lab part at home before coming to the lab. Make sure that you show your work to your TA and answer his questions to show that you know what you are doing before uploading your lab work and follow the instructions of your TAs.
4. In all labs if you are not told you may assume that inputs are correct.
5. In all labs when needed you have to provide a simple user interface for inputs and outputs.
6. Presentation of your work

   You have to provide a neat presentation prepared in txt form. Your programs must be easy to understand and well structured.

   Provide following six lines at the top of your submission for preliminary and lab work (make sure that you include the course no. CS224, important for ABET documentation).

   CS224
   Lab No.
   Section No.
   Your Full Name
   Bilkent ID
   Date

   Please also make sure that you work is identifiable: In terms of which program corresponds to which part of the lab.
7. **If we suspect that there is cheating we will send the work with the names of the students to the university disciplinary committee.**

DUE DATE PRELIMINARY WORK: SAME FOR ALL SECTIONS

No late submission will be accepted. Please do not try to break this rule and any other rule we set.

1. Please upload your programs of preliminary work to Moodle by 9:30 am on April 25, 2022 for similarity testing by MOSS. We plan to use MOSS both for problems solutions and program.

   Problems: Use the filename **StudentID_FirstName_LastName_SecNo_PRELIMproblem_LabNo.pdf** [A pdf fıle as its extension suggests, which contains your solutions to the Preliminary Part]. Only a pdf file is accepted. Any other form of submission receives 0 (zero).

Code: For the program part use the filename **StudentID_FirstName_LastName_SecNo_PRELIMcode_LabNo.txt** [A NOTEPAD FILE as its extension suggests, which contains only the program part Only a NOTEPAD FILE (txt file) is accepted. Any other form of submission receives 0 (zero).

2. Please note that the submission closes sharp at 9:30 am and no late submissions will be accepted. You can make resubmissions so do not wait for the last moment. Submit your work earlier and change your submitted work if necessary. Note that only the last submission will be graded.
3. Please familiarize yourself with the Moodle course interface, find the submission entry early, and avoid sending an email like "I cannot see the submission facility." (As of now it is not yet opened.)
4. Do not send your work by email attachment they will not be processed. They have to be in the Moodle system to be processed.

DUE DATE LAB WORK: (different for each section) YOUR LAB DAY

1. You have to demonstrate your lab work to your TA for grading. Do this by **12:00** in the morning lab and by **17:00** in the afternoon lab. Your TAs may give further instructions on this and they may make changes. If you wait idly and show your work last minute, your work may not be graded. Make sure that you follow your TA's instructions.
2. At the conclusion of the demo for getting your grade, you will **upload your Lab Work** to the Moodle Assignment, for similarity testing by MOSS. See lab part submission details below.
3. Aim to finish all of your lab work before coming to the lab, but make sure that you upload your work after making sure that your work is analyzed by your TA and/or you are given the permission by your TA to upload.
4. At the beginning of your submission files include the following make sure that each of them is in a separate line: Course No.: CS224, Lab No., Section No., Your Full Name, and your Bilkent ID.

Part 1. Preliminary Work (50 points)

You have to provide a neat presentation prepared by Word or a word processor with similar output quality. Handwritten answers will not be accepted.

1. (5 points: With 3 or more errors you get 0 points. Otherwise full point.) Fill in the empty cells of the following table. Assume that main memory size is 2 GB. **Index Size**: No. of bits needed to express the set number in an address, **Block Offset**: No. of bits needed to indicate the word offset in a block, **Byte Offset**: No. of bits needed to indicate the byte offset in a word. **Block Replacement Policy Needed**: Indicate if a block replacement policy such as FIFO, LRU, LFU (Least Frequently Used) etc. is needed (yes) or not (no). If some combinations are not possible mark them.

No.	Cache Size KB	N way cache	Word Size (no. of bits)	Block size (no. of words)	No. of Sets	Tag Size in bits	Index Size (Set No.) in bits	Block Offset Size in bits[1]	Byte Offset Size in bits[2]	Block Replacement Policy Needed (Yes/No)
1	64	1	32	4						
2	64	2	32	4						
3	64	4	32	8						
4	64	Full	32	8						
9	128	1	16	4						
10	128	2	16	4						
11	128	4	16	16						
12	128	Full	16	16						

[1] **Block Offset Size in bits:** Log_2(No. of words in a block)
[2] **Byte Offset Size in bits:** Log_2(No. of bytes in a word)

2. (5 points: With 3 or more errors you get 0 points. Otherwise full point.) Consider the following MIPS code segment. (Remember MIPS memory size is 4 GB.) Cache capacity is 16 words, Block size: 4 words, N= 2.

```
        addi   $t0, $0, 5
loop:   beq    $t0, $0, done
        lw     $t1, 0x24($0)
        lw     $t2, 0xAC($0)
        lw     $t3, 0xC8($0)
        addi   $t0, $t0, -1
        j      loop
done:
```

a. In the following table indicate the type of miss, if any: Compulsory, Conflict, Capacity.

Instruction	Iteration No.				
	1	2	3	4	5
lw $t1, 0x24($0)					
lw $t2, 0xAC($0)					
lw $t3, 0xC8($0)					

b. What is the total cache memory size in number of bits? Include the V bit your calculations. Show the details of your calculation.

c. State the number of AND and OR gates, EQUALITY COMPARATORs and MULTIPLEXERs needed to implement the cache memory. No drawing is needed.

3. (5 points: With 3 or more errors you get 0 points. Otherwise full point.) Consider the above MIPS code segment. Block size is 1 word. There is only 1 set. Cache memory size is 2 blocks. The block replacement policy is LRU.

a. In the following table indicate the type of miss, if any: Compulsory, Conflict, Capacity.

Instruction	**Iteration No.**				
	1	**2**	**3**	**4**	**5**
lw \$t1, 0x24(\$0)					
lw \$t2, 0xAC(\$0)					
lw \$t3, 0xC8(\$0)					

b. How many bits are needed for the implementation of LRU policy? What is the total cache memory size in number of bits? Include the V bit and the bit(s) used for LRU in your calculations. Show the details of your calculation.

c. State the number of AND and OR gates, EQUALITY COMPARATORs and MULTIPLEXERs needed to implement the cache memory. No drawing is needed.

4. (5 points, With 1 or more errors you get 0 points. Otherwise full point.) Consider a three level memory: L1 and L2 are for cache memory and the third level is for the main memory. Access time for L1 is 2 clock cycle, the access time for L2 is 4 clock cycles and main memory access time is 20 clock cycles. The miss rate for L1 is 10% and the miss rate for L2 is 5%. What is the effective clock cycle for memory access (AMAT in number of clock cycles)?

With 2 GHz clock rate how much time is needed for a program with 10^{10} instructions to execute?

5. (30 points) Write a program to copy the contents of a square matrix to another matrix of the same size. Provide a user interface for user interaction to demonstrate that your program is working properly. See below for the description of the user interface menu. Assume that in the main memory matrix elements are placed row by row. For example a 3 by 3 (N= 3) matrix would have the following values.

1	2	3
4	5	6
7	8	9

The row by row placement means that you will have the values of the above 3 x 3 matrix are stored as follows in the memory.

Matrix Index (Row No., Col. No.)	(1, 1)	(1, 2)	(1, 3)	(2, 1)	(2, 2)	(2, 3)	(3, 1)	(3, 2)	(3, 3)
Displacement With respect the beginning of the array containing the matrix	0	4	8	12	16	20	24	28	32
Value stored	1	2	3	4	5	6	7	8	9

In this configuration accessing the matrix element (i, j) simply involves computation of its displacement from the beginning of the array that stores the matrix elements. For example, the displacement of the matrix element with the index (i, j) with respect to the beginning of the array is (i - 1) x N x 4 + (j - 1) x 4, for a matrix of size N x N.

In your program for the matrix allocate an array with proper size using syscall code 9.

Your user interface must provide at least the following menu options:

1. Ask the user the matrix size in terms of its dimensions (N), and then ask the user enter matrix elements row by row.
2. Ask the user the matrix size in terms of its dimensions (N), and initialize the matrix entries with consecutive values (1, 2, 3 ...),
3. Display a desired element of the matrix by specifying its row and column number,
4. Display entire matrix row by row (for matrices of reasonable size, what size state in your user interface) please provide a good looking output,
5. Copy a matrix to another matrix: row wise or column wise (ask user).

2. Experiments with Data Cache Parameters (50 points)

Run your program with two reasonably large different matrix sizes that would provide meaningful observations.

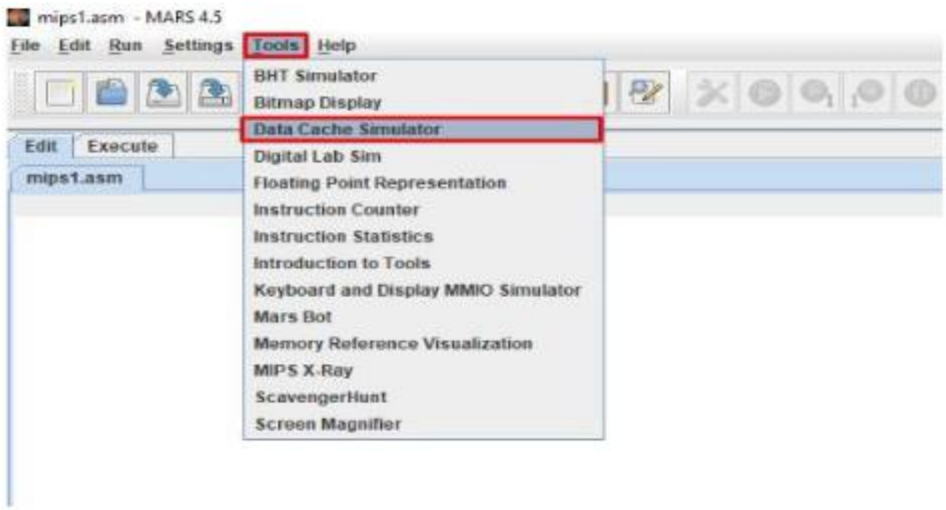
Report for Matrix Size 1: 25 Points

Report for Matrix Size 2: 25 Points

As described above make sure that you have a easy to follow presentation with numbered tables having proper heading etc.

a) **Direct Mapped Caches**: For the matrix sizes you have chosen, conduct tests with various cache sizes and block sizes, to determine the hit rate, miss rate and number of misses. Use at least 5 different cache sizes and 5 different block sizes (make sure your values are reasonable) in order to obtain curves like those of Figure 8.18 in the textbook (see below). Make a 5 x 5 table with your values, with miss rate and # of misses as the data at each row-column location. Make a graph of miss rate versus block size, parameterized by cache size, like Figure 8.18.

   Hint: You can reach the Cache Simulator from MARS/Tools/Data Cache Simulator as shown in the following image:



b) **Fully Associative Caches**: Pick 3 of your parameter points obtained in part for column-major copy a), one with good hit rate, one with medium hit rate, and one with poor hit rate. For these 3 results, there were 3 configuration pairs of cache size and block size that resulted in the data. Take the same 3 configuration pairs, but this time run the simulation with a fully associate cache, using LRU replacement policy. Compare the results obtained: the Direct Mapped good result versus the Fully Associative good result, the Direct Mapped medium result versus the Fully Associative medium result, and the Direct Mapped poor result versus the Fully Associative poor result. How much difference did the change to fully associative *architecture* make? Now change the replacement policy to Random replacement, and run the 3 tests again (using the same 3 configuration pairs). Does replacement policy make a significant difference? Record these 9 values in a new table, with 3 lines: for Direct Mapped, for Fully Associative-LRU and for Fully Associative-Random.

c) **N-way Set Associative Caches**: to save on hardware costs, fully set-associative caches are rarely used. Instead, most of the benefit can be obtained with an N-way set associative cache. Pick the medium hit rate configuration that you found in a) and used again in b), and change the architecture to N-way set associative. For several different set sizes (at least 4) and LRU replacement policy, run the program and record the hit rate, miss rate and number of misses. What set size gives the best result? How much improvement is gained as N (the number of blocks in a set) increases each step? Now repeat the tests, but for the good hit rate configuration from a) and b). Record these data and answer the same question again. Finally, repeat for the poor hit rate configuration.

Oral Interview with TA and Submission of your Data

Get ready for the interview with your TA, by gathering and analyzing your data, having it ready to submit in a clean understandable format. Be sure that you understand what you have done, and can interpret your data to the TA. Then call him over, and answer the questions he asks you.

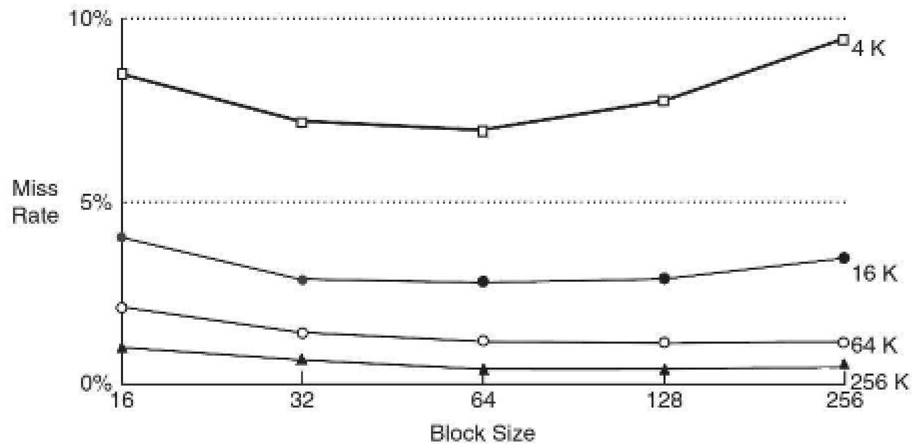


Figure 8.18 Miss rate versus block size and cache size on SPEC92 benchmark
Adapted with permission from Hennessy and Patterson, *Computer Architecture: A Quantitative Approach*, 5th ed., Morgan Kaufmann, 2012.

Part 3. Submit Lab Work for MOSS Similarity Testing

1. Submit your Lab Work MIPS codes for similarity testing to Moodle. See instructions below.
2. **Report**: Put your experiment results for Part 2 (including the tables and graphs) in a single PDF file. Use filename **StudentID_FirstName_LastName_SecNo_Lab6_lab_report.pdf** [pdf FILE as its extension suggests, which contains all the work done for the Lab Experiment Report Part].

   Code: Put your MIPS code for Part 1.5 into a .txt file. Use filename . Note that as you do the experiments your program may or may not involve changes. Whichever the case upload. **StudentID_FirstName_LastName_SecNo_Lab6_lab_code.txt** [A NOTEPAD FILE as its extension suggests, which contains the Program Code Part]
   Your program (code) will be compared against all the other programs in the class, by the MOSS program, to determine how similar it is (as an indication of plagiarism). So be sure that the code you submit is code that you actually wrote yourself! The same type of comparison is also planned for the reports.
3. *Even if you didn't finish, or didn't get the MIPS codes working, you must submit your code to the Moodle Assignment for similarity checking.*
4. Your codes will be compared against all the other codes in the class, by the MOSS program, to determine how similar it is (as an indication of plagiarism). So be sure that the code you submit is code that you actually wrote yourself!
5. For your preliminary and lab works to be graded you must attend the lab.

Part 4. Cleanup

1. After saving any files that you might want to have in the future to your own storage device, erase all the files you created from the computer in the lab.

2. When applicable put back all the hardware, boards, wires, tools, etc where they came from.
3. Clean up your lab desk, to leave it completely clean and ready for the next group who will come.

LAB POLICIES

1. You can do the lab only in your section. Missing your section time and doing in another day is not allowed.
2. The questions asked by the TA will have an effect on your lab score.
3. Lab score <u>will be reduced to 0</u> if the code is not submitted for similarity testing, or if it is plagiarized. MOSS-testing will be done, to determine similarity rates. Trivial changes to code will not hide plagiarism from MOSS—the algorithm is quite sophisticated and powerful. Please also note that obviously you should not use any program available on the web, or in a book, etc. since MOSS will find it. The use of the ideas we discussed in the classroom is not a problem.
4. You must be in lab, working on the lab, from the time lab starts until your work is finished and you leave.
5. No cell phone usage during lab.
6. Internet usage is permitted only to lab-related technical sites.