Assignment I

Problem Bank 21

Assignment Description:

The assignment aims to provide deeper understanding of cache by analysing it's behaviour using cache implementation of CPU- OS Simulator. The assignment has three parts.

- Part I deals with Cache Memory Management with Direct Mapping
- Part II deals with Cache Memory Management with Associative Mapping
- Part III deals with Cache Memory Management with Set Associative Mapping

- **Submission:** You will have to submit this documentation file and the name of the file should be GROUP-NUMBER.pdf. For Example, if your group number is 1, then the file name should be GROUP-1.pdf.

Submit the assignment by **26ᵗʰ December 2020, through canvas only**. File submitted by any means outside CANVAS will not be accepted and marked.

In case of any issues, please drop an email to the course TAs, Ms. Michelle Gonsalves (michelle.gonsalves@wilp.bits-pilani.ac.in).

Caution!!!

Assignments are designed for individual groups which may look similar and you may not notice minor changes in the assignments. Hence, refrain from copying or sharing documents with others. Any evidence of such practice will attract severe penalty.

Evaluation:

- The assignment carries 13 marks
- Grading will depend on
  - Contribution of each student in the implementation of the assignment
  - **Plagiarism or copying will result in -13 marks**

Assignment Set Number: 21

Group Name: 101

Contribution Table:

Contribution (This table should contain the list of all the students in the group. Clearly mention each student's contribution towards the assignment. Mention "No Contribution" in cases applicable.)

Sl. No.	Name (as appears in Canvas)	ID NO	Contribution
1	VIVEK BIRADAR	2020fc04271	100%
2	VIKRANT KUMAR SINGH	2020fc04272	100%
3	PATIL ATUL VILAS	2020fc04279	100%

Resource for Part I, II and III:

- Use following link to login to "eLearn" portal.
  - https://elearn.bits-pilani.ac.in
- Click on "My Virtual Lab – CSIS"
- Using your canvas credentials login in to Virtual lab
- In "BITS Pilani" Virtual lab click on "Resources". Click on "Computer Organization and software systems" course.
  - Use resources within "LabCapsule3: Cache Memory"

Code to be used:

The following code written in STL Language, implements searching of an element (key) in an array using linear search technique.

```
program LinearSearch
      var a array(50) byte

      writeln("Array Elements: ")

      for n = 0 to 10
            a(n) = n
            writeln(a(n))
      next
```

```
key = 10
writeln("Key to be searched: ",key)

found = 0

for n = 0 to 20
        temp = a(n)
        if temp = key then
                found = 1
                writeln("Key Found",temp)
                break
        end if
next
if found <> 1 then
        writeln("Key Not Found")
end if
end
```

General procedure to convert the given STL program in to ALP :

- Open CPU OS Simulator. Go to **advanced tab** and press **compiler** button
- Copy the above program in **Program Source** window
- Open **Compile** tab and press **compile** button
- In **Assembly Code,** enter **start address** and press **Load in Memory** button
- Now the assembly language program is available in CPU simulator.
- Set speed of execution to **FAST.**
- Open I/O console
- To run the program press **RUN** button.

**General Procedure to use Cache set up in CPU-OS simulator**

- After compiling and loading the assembly language code in CPU simulator, press "Cache-Pipeline" tab and select cache type as "data cache". Press "SHOW CACHE.." button.
- In the newly opened cache window, choose appropriate cache Type, cache size, set blocks, replacement algorithm and write back policy.

Part I: Direct Mapped Cache

a) Execute the above program by setting block size to 2, 4, 8, 16 and 32 for cache size =
8, 16 and 32. Record the observation in the following table.

Block Size	Cache size	# Hits	# Misses	% Miss Ratio	%Hit Ratio
2	8	124	62	33.33	66.67
4		114	72	38.71	61.29
8		91	95	51.07	48.92
2	16	138	48	25.8	74.2
4		148	38	20.43	79.56
8		141	45	24.19	75.81
16		146	40	21.50	78.49
2	32	143	43	23.11	76.88
4		160	26	13.97	86.02
8		172	14	7.52	92.47
16		176	10	5.37	94.62
32		180	6	3.22	96.77

b) Plot a single graph of Cache hit ratio Vs Block size with respect to cache size = 8, 16
and 32. Comment on the graph that is obtained.
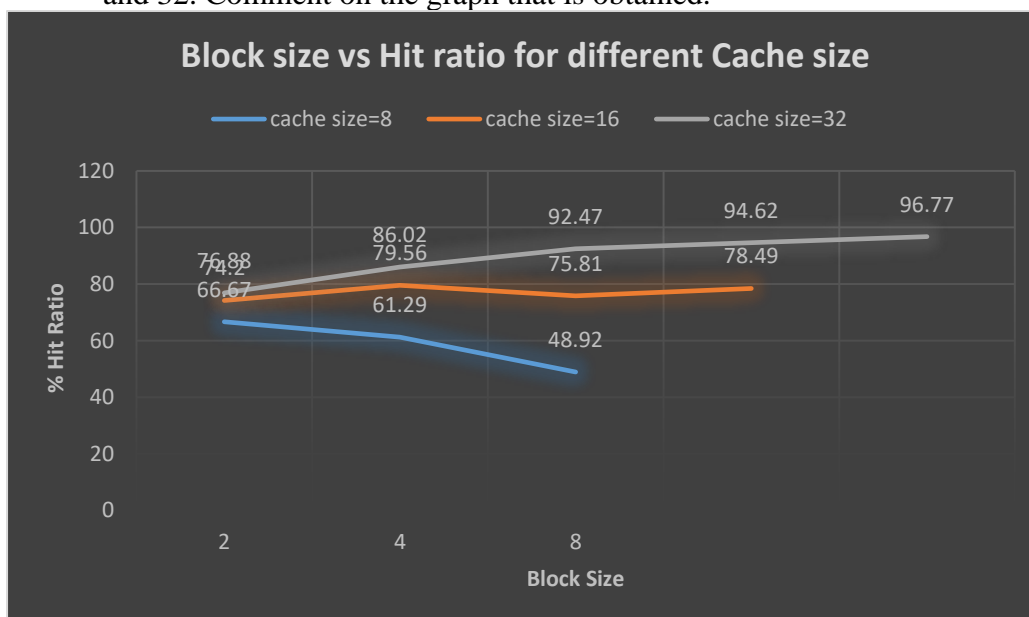
Explanation : From the readings, it can be observed that if we will increase the block size, hit ratio will also increase, it is because cache can use spatial locality if we will have larger block size.

c) Now, select cache type as "instruction cache". Fill in the following table and analyse the behaviour of Direct Mapped Cache. Which one is better with respect to Miss Ratio?

Block Size, Cache size	Miss	Hit	Miss Ratio
2, 8	170	158	0.518
2, 16	49	279	0.149
2, 32	30	298	0.0915

**Explanation :** With respect to miss ratio, for direct mapped cache block size=2 and cache size=32 is best.

Part II: Associative Mapped Cache

a) Execute the above program by setting block size to 2, 4, 8, 16 and 32 for cache size = 8, 16 and 32. Record the observation in the following table.

LRU Replacement Algorithm					
Block Size	Cache size	# Hits	# Misses	% Miss Ratio	%Hit Ratio
2	8	134	52	27.95	72.04
4		111	75	40.32	59.67
8		91	95	51.07	48.92
2	16	138	48	25.80	74.19
4		154	32	17.20	82.79
8		142	44	23.65	76.34

16		146	40	21.50	78.49
2	32	138	48	25.80	74.19
4		158	28	15.05	84.94
8		171	15	8.06	91.93
16		176	10	5.37	94.62
32		180	6	3.22	96.77

b) Plot a single graph of Cache hit ratio Vs Block size with respect to cache size = 8, 16 and 32. Comment on the graph that is obtained.
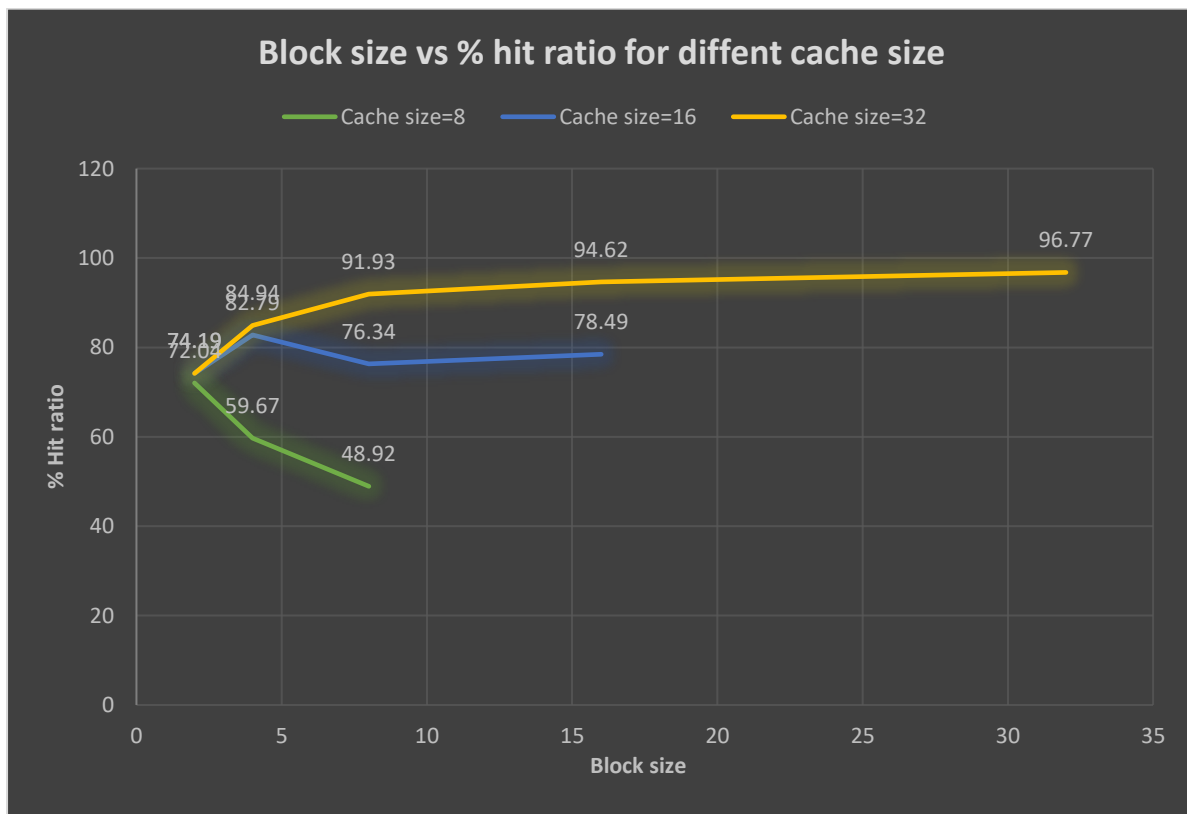
Explanation : From the readings, it can be observed that if we will increase the block size, hit ratio will also increase, it is because cache can use spatial locality if we will have larger block size.

d) Now, select cache type as "instruction cache". Fill in the following table and analyse the behaviour of associative mapped Cache. Which one is better with respect to Miss Ratio?

Block Size, Cache size	Miss	Hit	Miss Ratio
2, 8	180	148	0.548
2, 16	30	298	0.0915
2, 32	30	298	0.0915

Explanation : With respect to miss ratio, for direct mapped cache block size=2 and cache size=32 is best.

c) Fill up the following table for three different replacement algorithms and state which replacement algorithm is better and why ?

Replacement Algorithm : Random				
Block Size	Cache size	Miss	Hit	Hit ratio
2	4	97	89	0.4784
2	8	54	132	0.7096
2	16	47	139	0.7473
2	32	42	144	0.7741
2	64	40	146	0.7849
Replacement Algorithm : FIFO				
Block Size	Cache size	Miss	Hit	Hit ratio
2	4	94	92	0.4946
2	8	52	134	0.7204
2	16	48	138	0.7419
2	32	48	138	0.7419
2	64	40	146	0.7849
Replacement Algorithm : LRU				
Block Size	Cache size	Miss	Hit	Hit ratio
2	4	103	83	0.4462
2	8	48	138	0.7419
2	16	48	138	0.7419
2	32	46	140	0.7526
2	64	40	146	0.7849

e) Plot the graph of Cache Hit Ratio Vs Cache size with respect to different replacement algorithms. Comment on the graph that is obtained.

Cache size vs Hit ratio for different algorithm
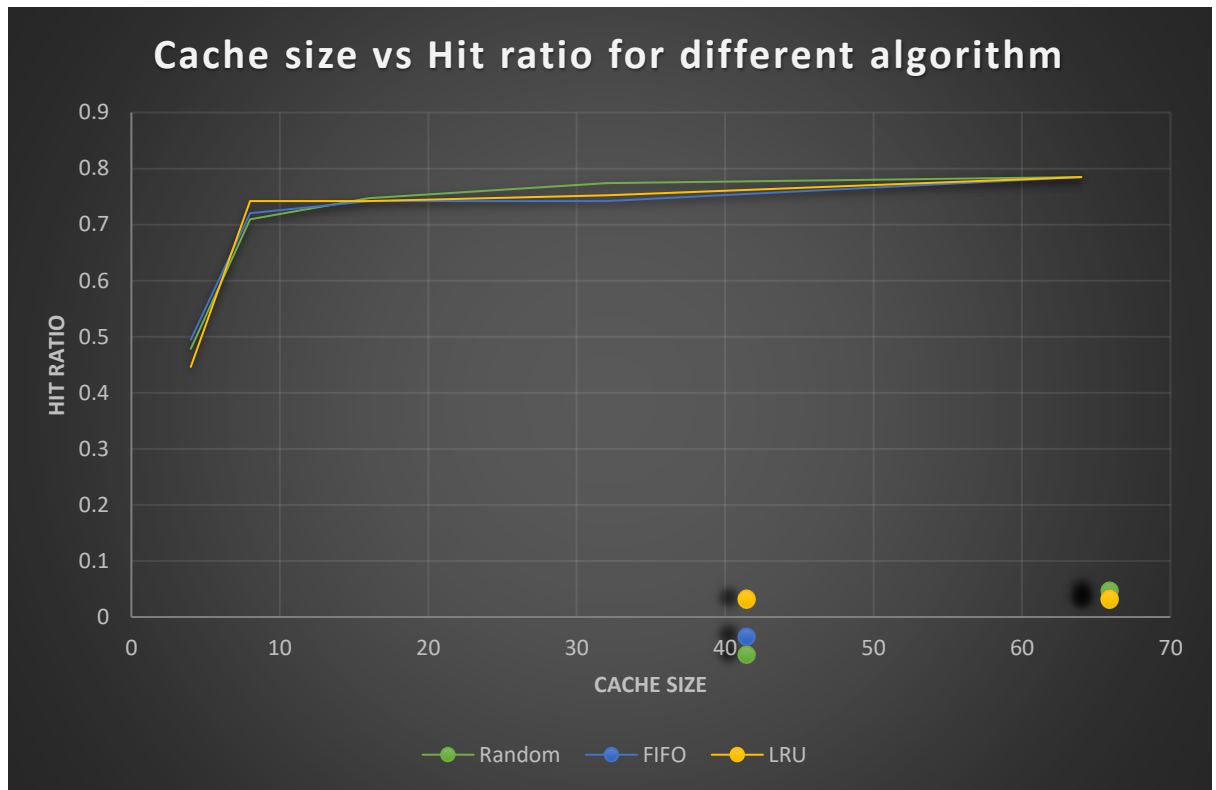
Explanation : From the readings, it can be observed that if we will increase the cache size, hit ratio will also increase.

Part III: Set Associative Mapped Cache

Execute the above program by setting the following Parameters:
- Number of sets (Set Blocks) : 2 way
- Cache Type : Set Associative
- Replacement: LRU/FIFO/Random

a) Fill up the following table for three different replacement algorithms and state which replacement algorithm is better and why ?
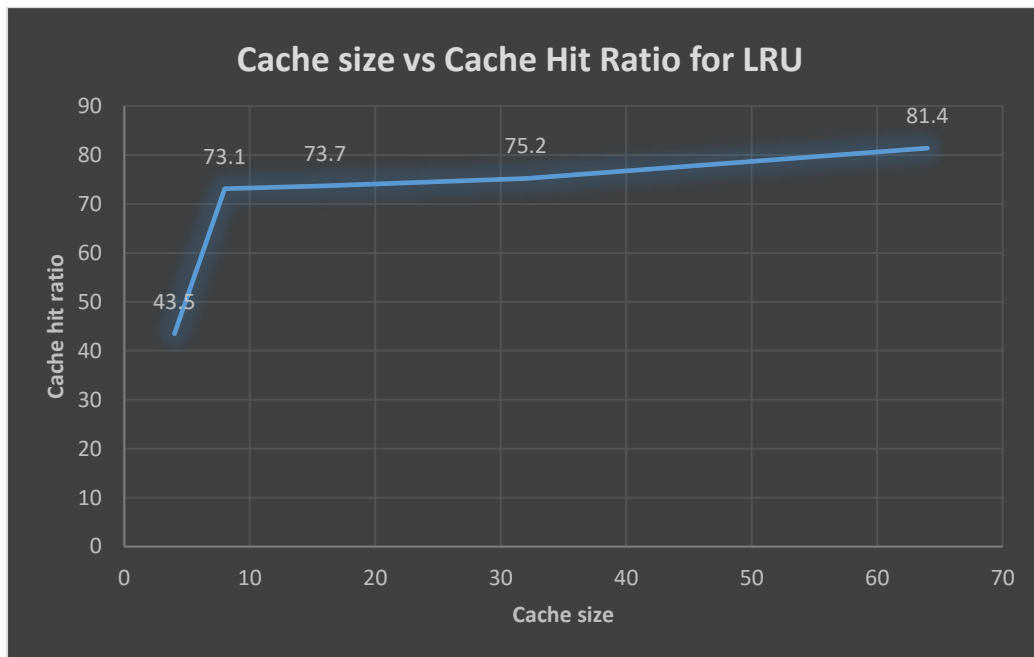
Replacement Algorithm : Random				
Block Size	Cache size	Miss	Hit	Hit ratio
2	4	94	88	49.4
2	8	58	131	69.4
2	16	49	133	73.1
2	32	50	139	73.6
2	64	34	148	81.4
Replacement Algorithm : FIFO				
Block Size	Cache size	Miss	Hit	Hit ratio
2	4	97	92	48.7
2	8	52	130	71.5

2	16	51	138	73.1
2	32	44	138	75.9
2	64	43	146	77.3
Replacement Algorithm : LRU				
Block Size	Cache size	Miss	Hit	Hit ratio
2	4	103	79	43.5
2	8	51	138	73.1
2	16	48	134	73.7
2	32	47	142	75.2
2	64	34	148	81.4

c) Plot the graph of Cache Hit Ratio Vs Cache size with respect to different replacement algorithms. Comment on the graph that is obtained.

Cache size vs Cache Hit Ratio for LRU
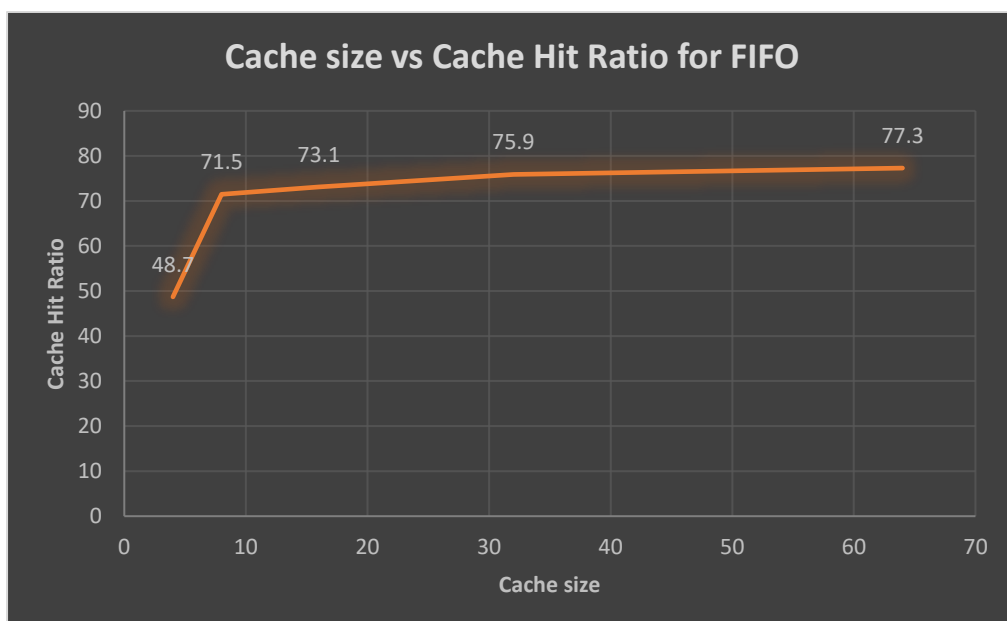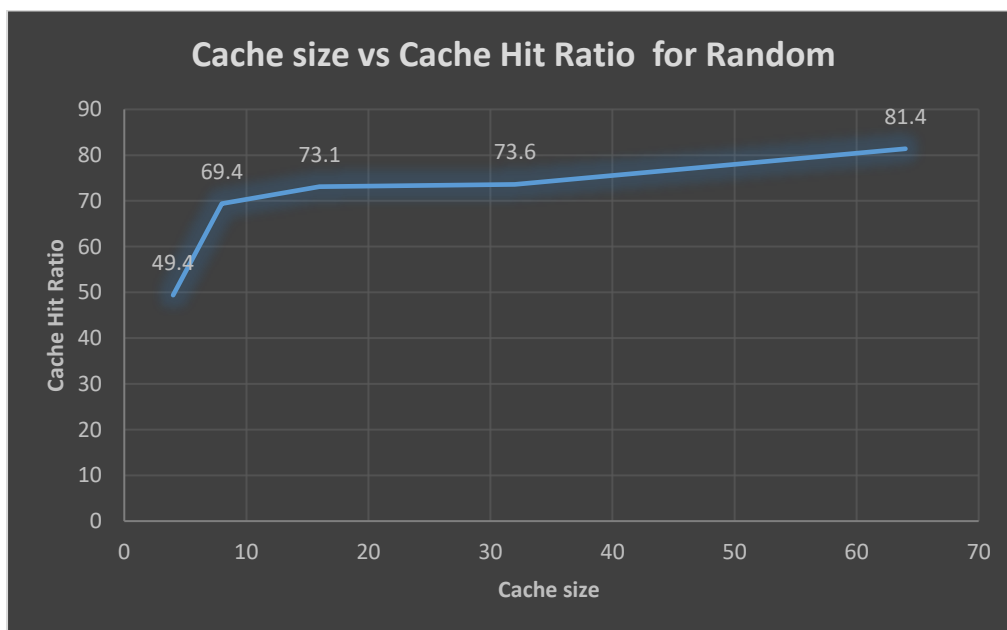
Explanation : From the readings, it can be observed that if we will increase the cache size, hit ratio will also increase,

c) Fill in the following table and analyse the behaviour of Set Associate Cache. Which one is better and why?

Replacement Algorithm : LRU				
Block Size, Cache size	Set Blocks	Miss	Hit	Hit ratio
2, 64	2 – Way	43	146	77.3
2, 64	4 – Way	40	146	79.5
2, 64	8 – Way	40	146	79.5

Explanation : From the readings we can observe that 4-way and 8-way has similar hit ratio, so we can say 4-way is best to use from given options. Costing for 4-way will be less than 8-way as well. It is because 4-way will require less number of bits in comparator to reach specific set address.