Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 4/4/2021

Name: Suhan B Revankar	SRN: PES2UG19CS412	Section G

Week#____9_____ Program Number:_____1__

Consider a direct mapped cache of size 16 bytes with block size 4 bytes. The size of main memory is 256 bytes. Find Number of bits in tag, index and offset. The processor generates requests as follows
1,4,8,5,14,11,13,38,9,B,4,2B,5,6,9,11.
Find hit rate and miss rate.

a) Cache Address Table showing the split-up of the address fields for the requests generated by the processor

HEX	DEC	TAG	INDEX	OFFSET
1	1	000	000000	00000001
4	4	000	000000	00000100
8	8	000	000000	00001000
5	5	000	000000	00000101
14	20	000	000000	00010100
11	17	000	000000	00010001
13	19	000	000000	00010011
38	56	000	000000	00111000
9	9	000	000000	00001001
B	11	000	000000	00001011
4	4	000	000000	00000100
2B	43	000	000000	00101011
5	5	000	000000	00000101
6	6	000	000000	00000110
9	9	000	000000	00001001
11	17	000	000000	00010001

+3 Instruction Breakdown

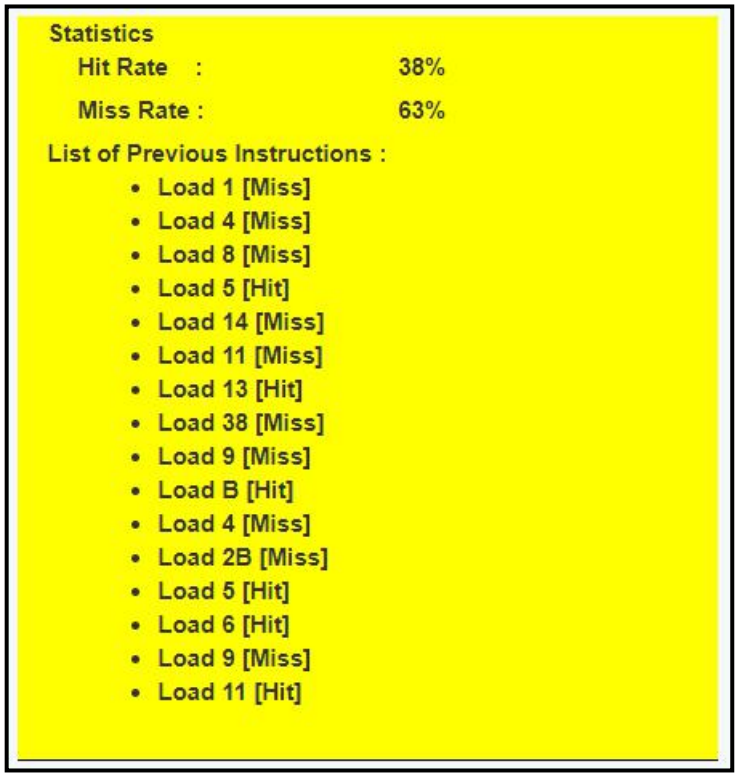
TAG	INDEX	**OFFSET**
4 bil	2 bit	2 bit

b) Screenshot showing the Cache Table

Cache Table

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	0001	BLOCK 4 WORD 0 - 3	0
1	1	0000	BLOCK 1 WORD 0 - 3	0
2	1	0000	BLOCK 2 WORD 0 - 3	0
3	0	-	0	0

c) Screenshot showing hit and miss rates



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 4/4/2021

Name: Suhan B Revankar	SRN: PES2UG19CS412	Section G

Week#____9_____ Program Number:_____2___

Consider a direct mapped cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB. Find Number of bits in tag. Randomly generate 10 addresses and find hit rate and miss rate.

a) Cache Address Table showing the splitup of the address fields for the requests generated by the processor

HEX	DEC	TAG	INDEX	OFFSET
1	1	000	000000	00000001
4	4	000	000000	00000100
8	8	000	000000	00001000
5	5	000	000000	00000101
14	20	000	000000	00010100
11	17	000	000000	00010001
13	19	000	000000	00010011
38	56	000	000000	00111000
9	9	000	000000	00001001
B	11	000	000000	00001011
4	4	000	000000	00000100
2B	43	000	000000	00101011
5	5	000	000000	00000101
6	6	000	000000	00000110
9	9	000	000000	00001001
11	17	000	000000	00010001

43 Instruction Breakdown

TAG	INDEX	**OFFSET**
3 bit	6 bit	8 bit

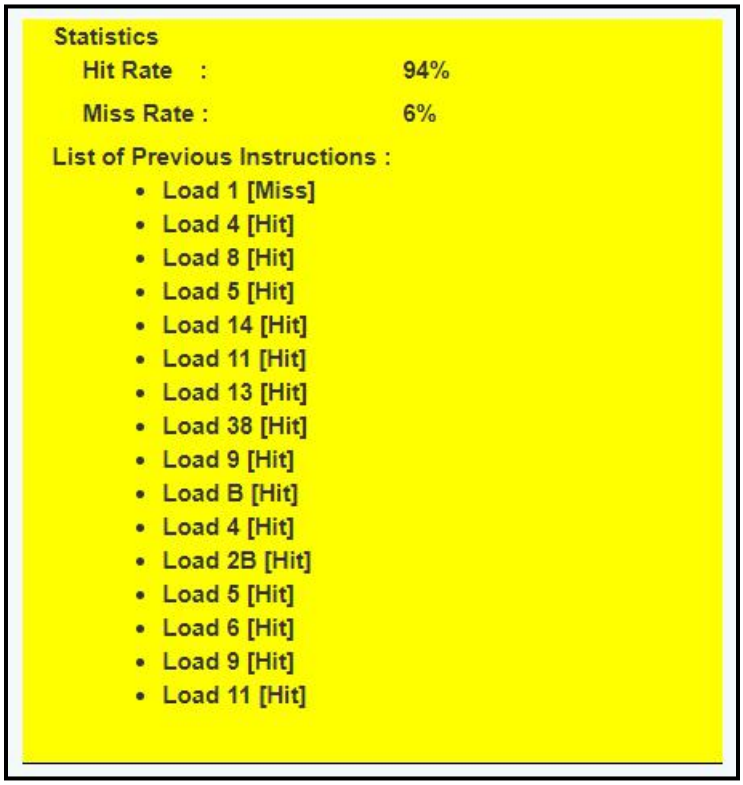
b) Screenshot showing the Cache Table

Cache Table

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	000	BLOCK 0 WORD 0 - 255	0
1	0	-	0	0
2	0	-	0	0
3	0	-	0	0
4	0	-	0	0
5	0	-	0	0
6	0	-	0	0
7	0	-	0	0
8	0	-	0	0
9	0	-	0	0
10	0	-	0	0
11	0	-	0	0
12	0	-	0	0
13	0	-	0	0
14	0	-	0	0
15	0	-	0	0
16	0	-	0	0
17	0	-	0	0
18	0	-	0	0
19	0	-	0	0
20	0	-	0	0
21	0	-	0	0
22	0	-	0	0
23	0	-	0	0
24	0	-	0	0
25	0	-	0	0
26	0	-	0	0
27	0	-	0	0
28	0	-	0	0
29	0	-	0	0
30	0	-	0	0
31	0	-	0	0
32	0	-	0	0
33	0	-	0	0
34	0	-	0	0
35	0	-	0	0
36	0	-	0	0
37	0	-	0	0
38	0	-	0	0
39	0	-	0	0
40	0	-	0	0
41	0	-	0	0
42	0	-	0	0
43	0	-	0	0
44	0	-	0	0
45	0	-	0	0
46	0	-	0	0
47	0	-	0	0

48	0		0	0
49	0		0	0
50	0		0	0
51	0		0	0
52	0		0	0
53	0		0	0
54	0		0	0
55	0		0	0
56	0		0	0
57	0		0	0
58	0		0	0
59	0		0	0
60	0		0	0
61	0		0	0
62	0		0	0
63	0		0	0

c) Screenshot showing hit and miss rates



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 4/4/2021

Name: Suhan B Revankar	SRN: PES2UG19CS412	Section G

Week#____9_____ Program Number:_____3___

A computer system uses 16-bit memory addresses. It has a 2K-byte cache organized in a direct-mapped manner with 64 bytes per cache block. Assume that the size of each memory word is 1 byte.

a) Calculate the number of bits in each of the Tag, Block, and Word fields of the memory address.

Cache size = 2048 B = 2^{11}

Block size = 64 B = 2^6 = 6 bits = Word bits

$\therefore$ the block bits are = $\frac{2^{11}}{2^6} = 2^{11-6}$ = 5 bits

$\Rightarrow$ Tag bits = $16 - (5 + 6) = 16 - 11$ = 5 bits

b) When a program is executed, the processor reads data sequentially from the following word addresses: 128, 144, 2176, 2180, 128, 2176

128: 00000 00010 000000 → Miss

144: 00000 00010 010000 → Hit

2176: 00001 00010 000000 → Miss

2180: 00001 00010 000100 → Hit

128: 00000 00010 000000 → Miss

2176: 00001 00010 000000 → Miss

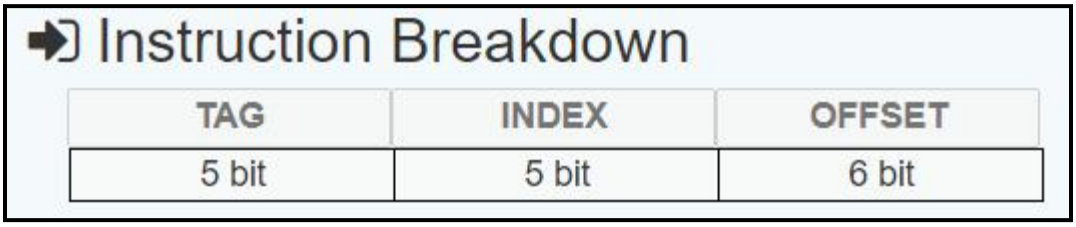
Miss rate = 3/6 = 0.67 or 67%

Hit rate = 2/6 = 0.33 or 33%

All the above addresses are shown in decimal values. Assume that the cache is initially empty. For each of the above addresses, indicate whether the cache access will result in a hit or a miss.

a) Cache Address Table showing the splitup of the address fields for the requests generated by the processor

HEX	DEC	TAG	INDEX	OFFSET
80	128	00000	00010	000000
90	144	00000	00010	010000
880	2176	00001	00010	000000
884	2180	00001	00010	000100
80	128	00000	00010	000000
880	2176	00001	00010	000000

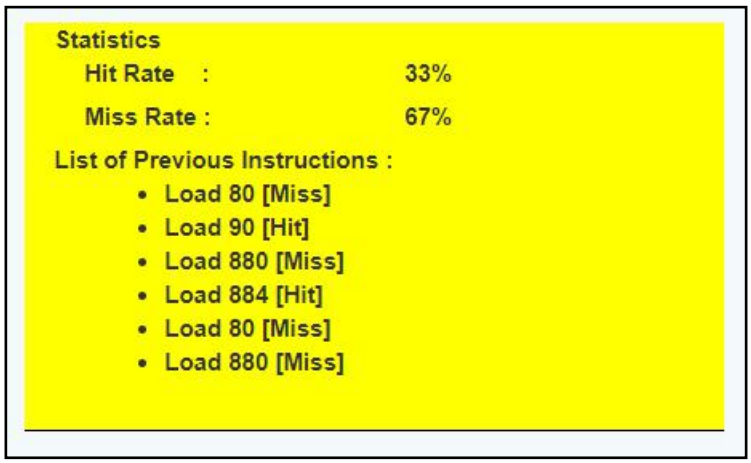


b) Screenshot showing the Cache Table

Cache Table

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	0	-	0	0
1	0	-	0	0
2	1	00001	BLOCK 22 WORD 0 - 63	0
3	0	-	0	0
4	0	-	0	0
5	0	-	0	0
6	0	-	0	0
7	0	-	0	0
8	0	-	0	0
9	0	-	0	0
10	0	-	0	0
11	0	-	0	0
12	0	-	0	0
13	0	-	0	0
14	0	-	0	0
15	0	-	0	0
16	0	-	0	0
17	0	-	0	0
18	0	-	0	0
19	0	-	0	0
20	0	-	0	0
21	0	-	0	0
22	0	-	0	0
23	0	-	0	0
24	0	-	0	0
25	0	-	0	0
26	0	-	0	0
27	0	-	0	0
28	0	-	0	0
29	0	-	0	0
30	0	-	0	0
31	0	-	0	0

c) Screenshot showing hit and miss rates



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 4/4/2021

Name: Suhan B Revankar	SRN: PES2UG19CS412	Section G

Week#____9_____ Program Number:___4__

Consider a 2-way set associative cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB. Find Number of bits in tag. Randomly generate 10 addresses and find hit rate and miss rate.

a) Cache Address Table showing the splitup of the address fields for the requests generated by the processor

HEX	DEC	TAG	INDEX	OFFSET
1	1	000	000000	00000001
4	4	000	000000	00000100
8	8	000	000000	00001000
5	5	000	000000	00000101
14	20	000	000000	00010100
11	17	000	000000	00010001
13	19	000	000000	00010011
38	56	000	000000	00111000
9	9	000	000000	00001001
B	11	000	000000	00001011
4	4	000	000000	00000100
2B	43	000	000000	00101011
5	5	000	000000	00000101
6	6	000	000000	00000110
9	9	000	000000	00001001
11	17	000	000000	00010001

43 Instruction Breakdown

TAG	INDEX	**OFFSET**
4 bit	5 bit	8 bit

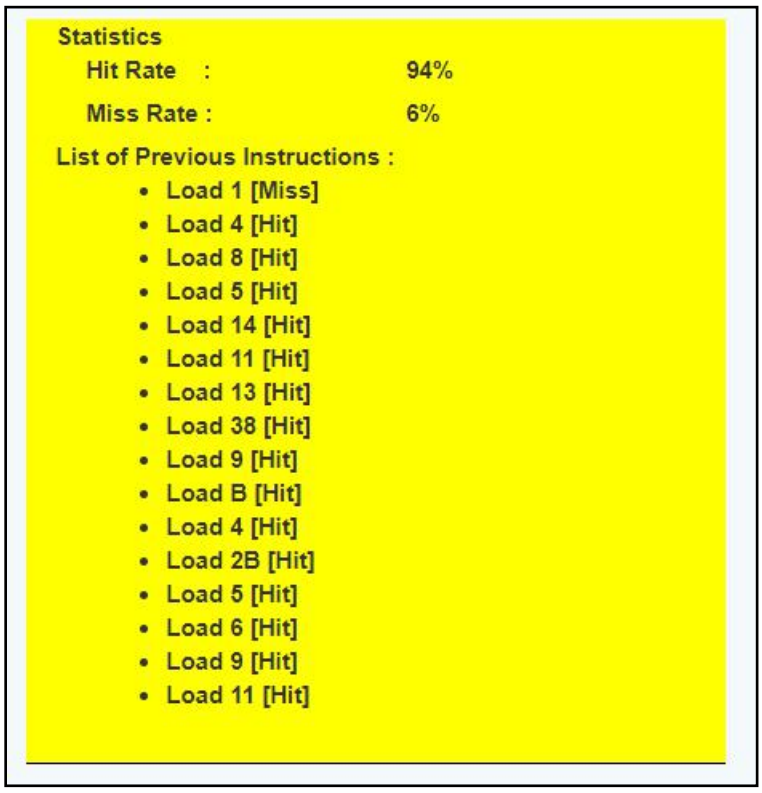
b) Screenshot showing the Cache Table

Cache Table

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	0	B. 0 W. 0 - 255	0
1	0	-	0	0
2	0	-	0	0
3	0	-	0	0
4	0	-	0	0
5	0	-	0	0
6	0	-	0	0
7	0	-	0	0
8	0	-	0	0
9	0	-	0	0
10	0	-	0	0
11	0	-	0	0
12	0	-	0	0
13	0	-	0	0
14	0	-	0	0
15	0	-	0	0
16	0	-	0	0
17	0	-	0	0
18	0	-	0	0
19	0	-	0	0
20	0	-	0	0
21	0	-	0	0
22	0	-	0	0
23	0	-	0	0

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	0	-	0	0
1	0	-	0	0
2	0	-	0	0
3	0	-	0	0
4	0	-	0	0
5	0	-	0	0
6	0	-	0	0
7	0	-	0	0
8	0	-	0	0
9	0	-	0	0
10	0	-	0	0
11	0	-	0	0
12	0	-	0	0
13	0	-	0	0
14	0	-	0	0
15	0	-	0	0
16	0	-	0	0
17	0	-	0	0
18	0	-	0	0
19	0	-	0	0
20	0	-	0	0
21	0	-	0	0
22	0	-	0	0
23	0	-	0	0

c) Screenshot showing hit and miss rates



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 4/4/2021

Name: Suhan B Revankar	SRN: PES2UG19CS412	Section G

Week#____9_____ Program Number:___5__

Consider a main memory having 64 byte capacity and cache memory of 8 bytes initially empty .The following addresses are generated by the CPU. All values in hexadecimal. Clearly label data that is replaced in cache lines Show the cache memory table and filled data in the cache lines of block size 1 byte. LRU Policy is used.

The cache is mapped as

a) Direct Mapped

b) Two way set Associative

c) Four Way Set associative

d) Fully Associative

a) Cache Address Table showing the splitup of the address fields for the requests generated by the processor

HEX	DEC	TAG	INDEX	OFFSET
1	1	000	000000	00000001
4	4	000	000000	00000100
8	8	000	000000	00001000
5	5	000	000000	00000101
14	20	000	000000	00010100
11	17	000	000000	00010001
13	19	000	000000	00010011
38	56	000	000000	00111000
9	9	000	000000	00001001
B	11	000	000000	00001011
4	4	000	000000	00000100
2B	43	000	000000	00101011
5	5	000	000000	00000101
6	6	000	000000	00000110
9	9	000	000000	00001001
11	17	000	000000	00010001

i) Direct mapping

Instruction Breakdown

TAG	INDEX	OFFSET
3 bit	2 bit	1 bit

ii) 2-way set associative

Instruction Breakdown

TAG	INDEX	OFFSET
4 bit	1 bit	1 bit

iii) 4-way set associative

Instruction Breakdown

TAG	INDEX	OFFSET
5 bit	0 bit	1 bit

iv) Fully associative

Instruction Breakdown

BLOCK	OFFSET
5 bit	1 bit

b) Screenshot showing the Cache Table

i) Direct mapping

Cache Table

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	010	BLOCK 8 WORD 0 - 1	0
1	1	101	BLOCK 15 WORD 0 - 1	0
2	1	000	BLOCK 2 WORD 0 - 1	0
3	1	000	BLOCK 3 WORD 0 - 1	0

ii) 2-way set associative

Cache Table

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	4	BLOCK 8 WORD 0 - 1	0
1	1	a	BLOCK 15 WORD 0 - 1	0

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	2	BLOCK 4 WORD 0 - 1	0
1	1	1	BLOCK 3 WORD 0 - 1	0

iii) 4-way set associative

Cache Table

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	4	B. 4 W. 0 - 1	0

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	8	B. 8 W. 0 - 1	0

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	2	B. 2 W. 0 - 1	0

Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	3	B. 3 W. 0 - 1	0

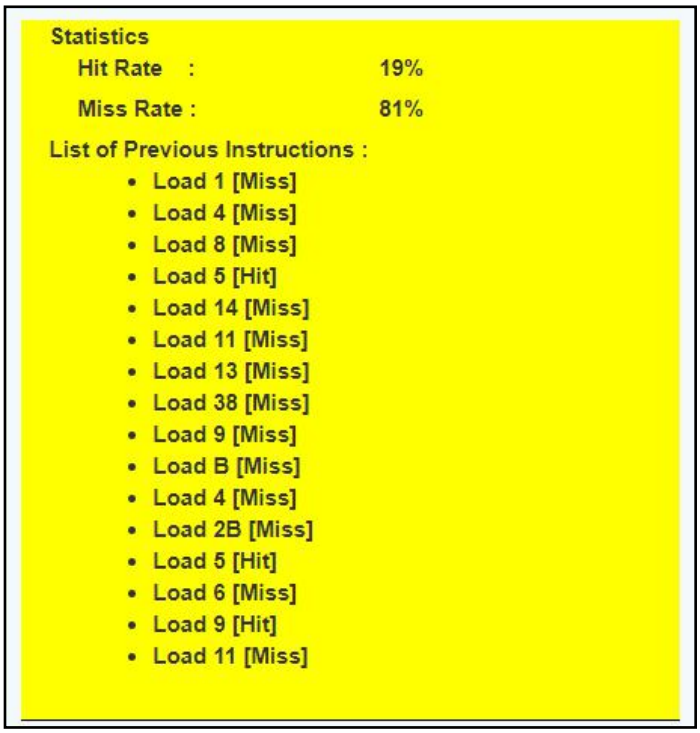
iv) Fully associative

Cache Table

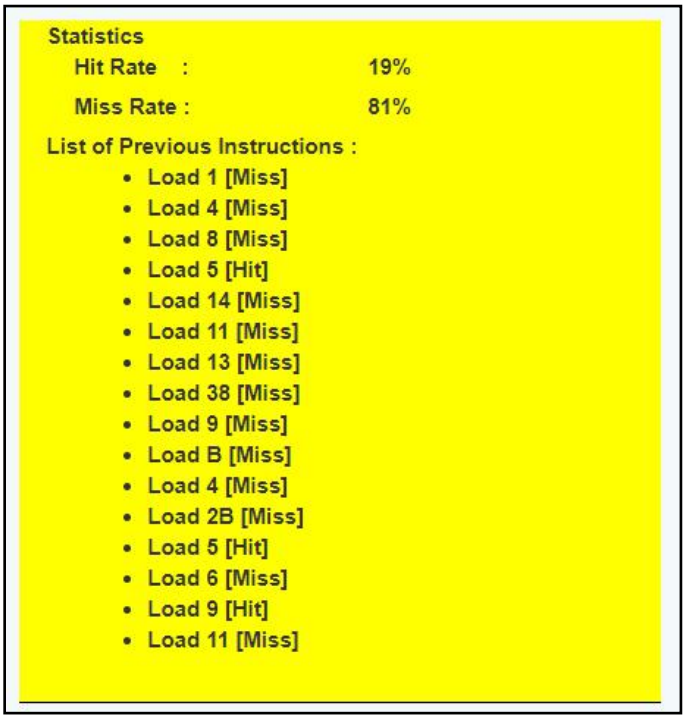
Index	Valid	Tag	Data (Hex)	Dirty Bit
0	1	00100	BLOCK 4 WORD 0 - 1	0
1	1	01000	BLOCK 8 WORD 0 - 1	0
2	1	00010	BLOCK 2 WORD 0 - 1	0
3	1	00011	BLOCK 3 WORD 0 - 1	0

c) Screenshot showing hit and miss rates

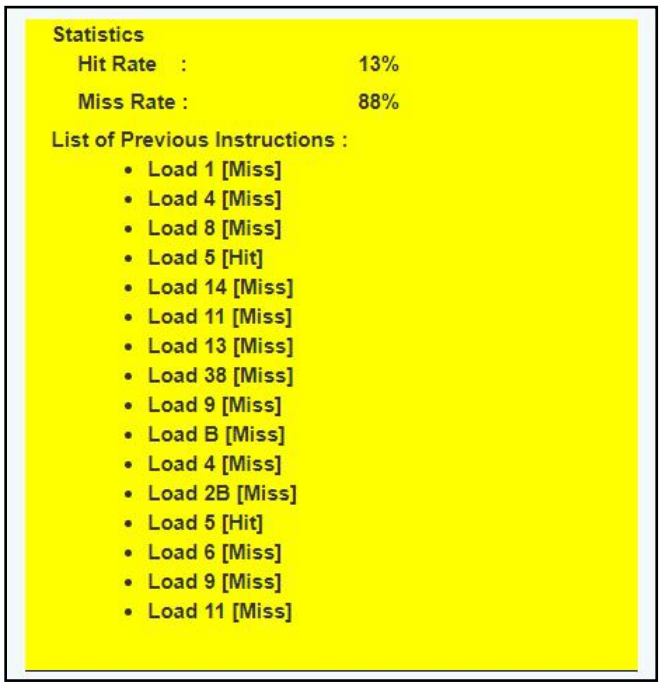
i) Direct mapping



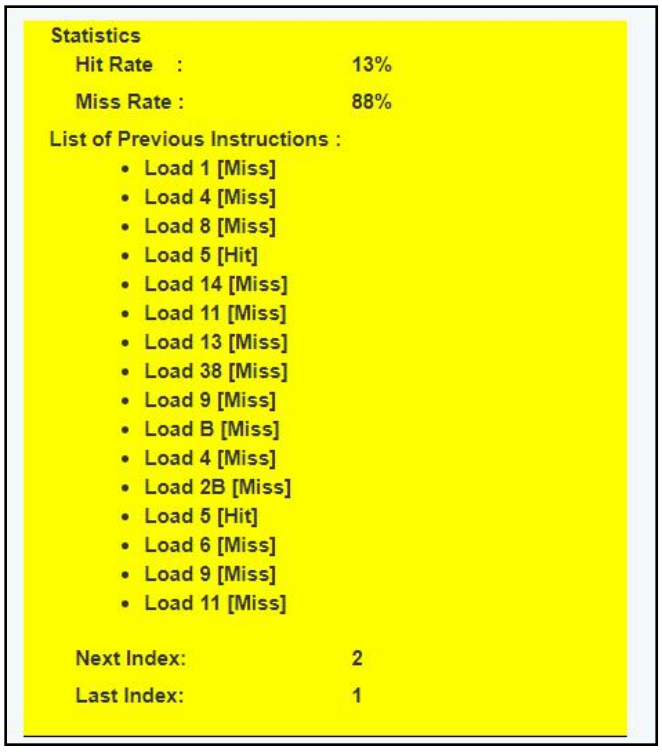
ii) 2-way set associative



iii) 4-way set associative



iv) Fully associative



Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature:Suhan B Revankar
Name: Suhan B
Revankar
SRN: PES2UG19CS412
Section: G
Date: 4/4/2021