

Lab Sheet-Direct Mapped Cache

1. Objective:

At the end of this lab session, the student should be able to:

- Understand direct mapped cache
- Understand the behaviour of direct-mapped cache by varying block size and cache size

2. Theory:

Cache memory is a small amount of fast memory placed between the CPU and the main memory to bridge the gap in access times. Due to the locality of memory references, the cache memory can enhance the computer system's performance. The efficiency gained by using a cache memory varies depending on cache size, block size, and other cache parameters, but it also depends on the program and data.

The mapping function is the way of associating memory blocks with the cache line. Three mapping functions available are

- Direct mapped
- Associative
- Set associative

The direct mapping technique is the simplest way of associating main memory blocks with the cache lines. In this technique, block k of the main memory maps into block k modulo m of the cache, where m is the total number of blocks. Since more than one main memory block is mapped onto a given cache block position, contention may arise for that position. This situation may occur even when the cache is not full. Conflict is resolved by allowing the new block to overwrite the current resident block. So the replacement algorithm is trivial.

3. Steps to perform :

. The following program should be used for the analysis.

```
LDB 00, R00  
LDB 01, R01
```

LDB 02, R02
LDB 03, R03
LDB 04, R04
LDB 05, R05
LDB 06, R06
LDB 07, R07
LDB 08, R08
LDB 09, R09
LDB 10, R10
LDB 11, R11
LDB 12, R12
LDB 13, R13
HLT

The procedure to enter the above code into the simulator is as follows:

- Create a program name and enter a base address in the **Base Address** text box
- Enter CPU instructions in the program
- Press the “Cache-Pipeline” tab and select the cache type as “data cache”, as shown in Figure 1. Press the “SHOW CACHE...” button. A new window will be opened, as shown in Figure 2.

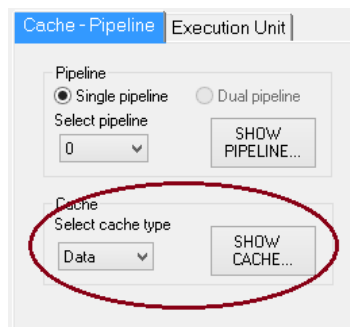


Figure 1: Cache – Pipeline setting

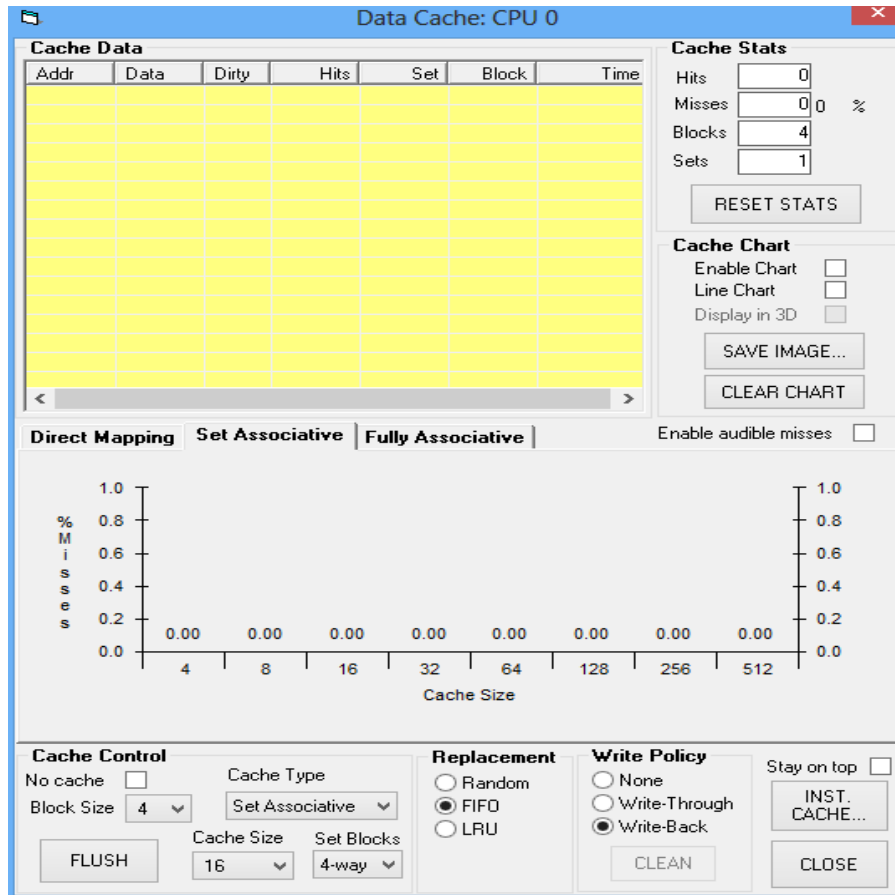


Figure 2: Data cache window

- (d) Set cache type to Direct mapped cache. Also, set block size, cache size and write policy.
- (e) Execute the program.
- (f) Note down cache miss, hit and compute the hit ratio.

Problem No 1:

Step 1: Set the following parameters in the cache control block of the data cache window, as shown in Figure 2.

- Block Size: 2
- Cache Size: 16 bytes
- Number of Blocks: 8 (automatically updated, which is equal to Cache size / Block Size)
- Cache Type: Direct Mapped
- Select Enable Chart check box
- Select stay on top check box

Step 2: Run the program in a single step and observe the changes on the data cache window

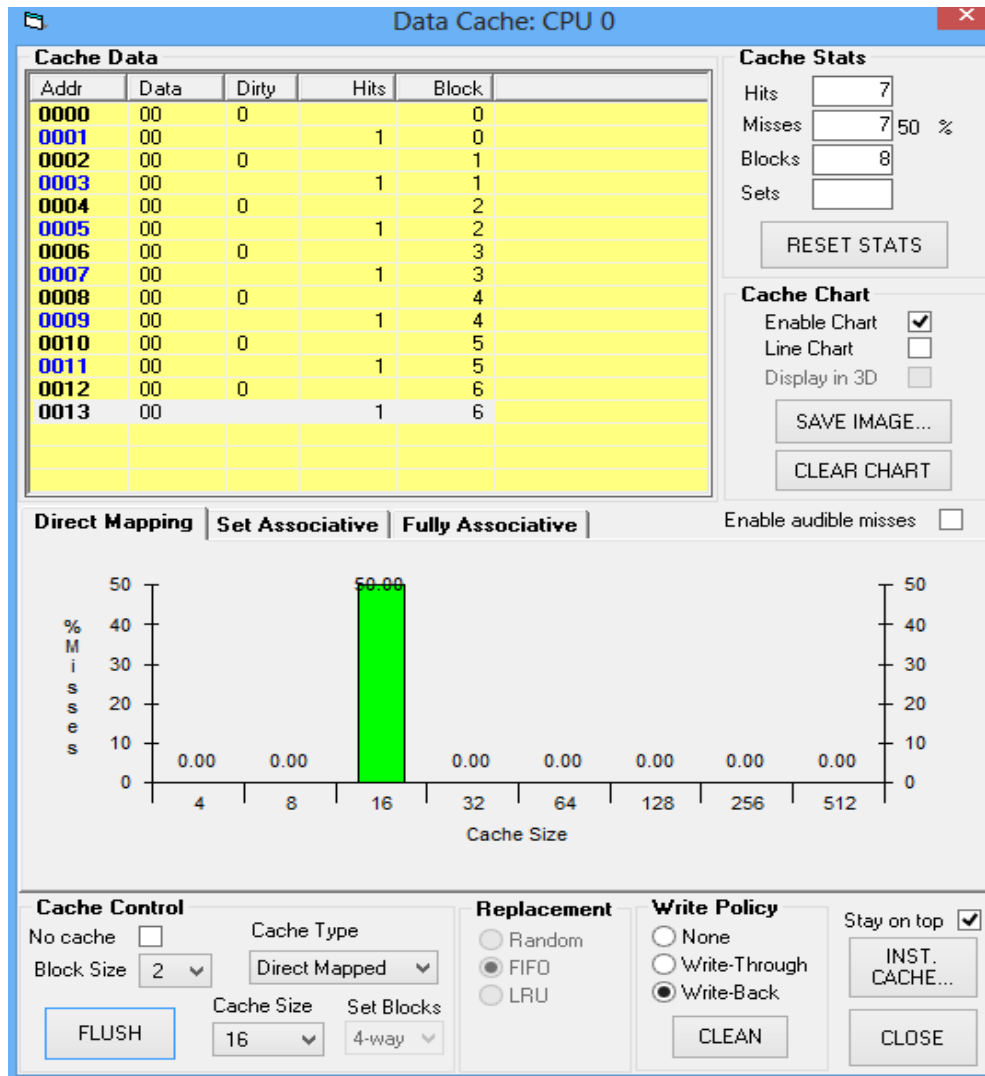


Figure 3: Data Cache window for problem no. 1

Step 3: Note down the number of hits, misses and hit ratio

Problem No 2: Analysis of direct-mapped cache by varying block size.

Set the following parameters:

- Blocks: 8
- Cache Type: Direct Mapped
- Cache Size: 16 bytes
- Select Enable Chart check box
- Select stay on top check box

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

Block Size	Number of blocks	Miss	Hit	Miss ratio
2	16	7	7	.5
4	16	4	10	0.285

8	16	2	12	0.142
16	16	1	13	0.0714
32	16	1	13	0.0714

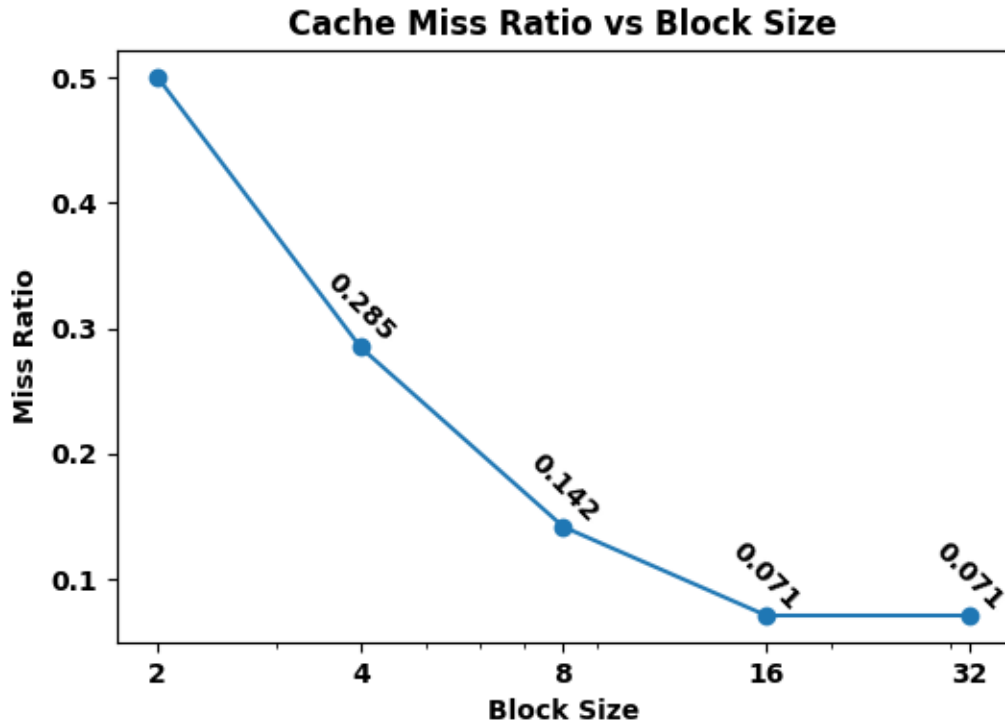


Figure 4: Graph of Cache miss ratio Vs Block size for problem no. 2

Problem No 3: Analysis of direct-mapped cache by varying cache size.

Set the following parameters:

- Block size: 2
- Cache Type: Direct Mapped
- Select Enable Chart check box
- Select stay on top check box

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

Block Size	Cache size	Miss	Hit	Miss ratio
2	4	7	7	0.5
2	8	7	7	0.5
2	16	7	7	0.5
2	32	7	7	0.5
2	64	7	7	0.5

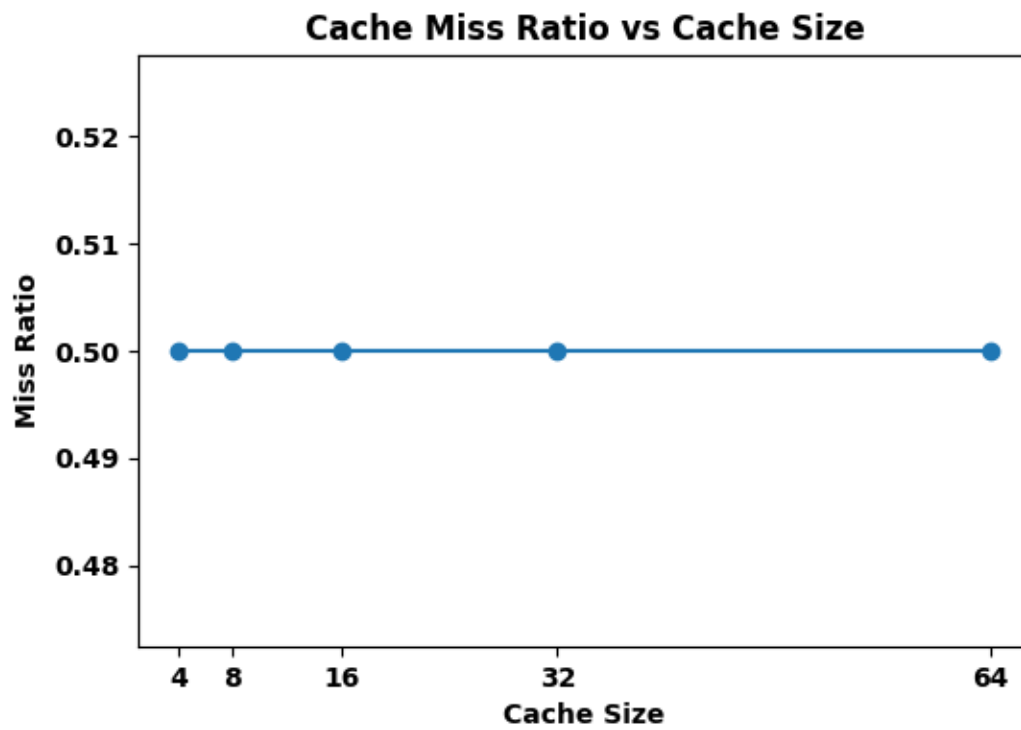


Figure 5: Graph of Cache miss ratio Vs Cache size for problem no. 3