**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

Date: 24/03/23

|  |  |  |
| --- | --- | --- |
| Name: Nikhil Girish | SRN: PES2UG21CS334 | Section: F |

Week#\_\_\_\_8\_\_\_\_\_\_ Number: \_\_\_\_1\_\_

Consider a direct mapped cache of size 16 bytes with block size 4 bytes. The size of main memory is 256 bytes.

a)Find Number of bits in tag, index and offset.

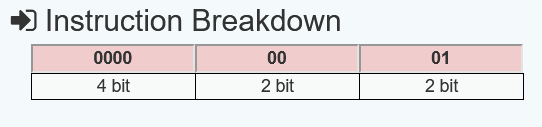
* Tag = 6 bits
* Index = 2 bits
* Offset = 2 bits

b)The processor generates requests as follows

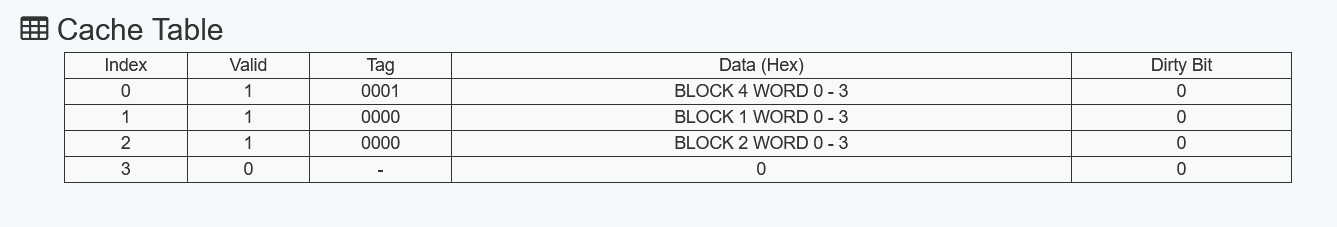
1,4,8,5,14,11,13,38,9,B,4,2B,5,6,9,11.

Find hit rate and miss rate.

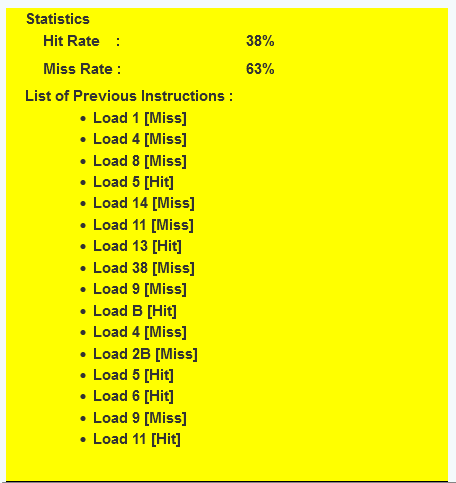
1. Cache Address Table showing the splitup of the address fields for the requests generated by the processor



1. Screenshot showing the Cache Table



1. Screenshot showing hit and miss rates



**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

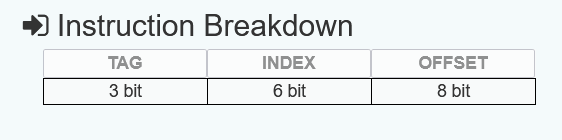
Date: 24/03/23

|  |  |  |
| --- | --- | --- |
| Name: Nikhil Girish | SRN: PES2UG21CS334 | Section: F |

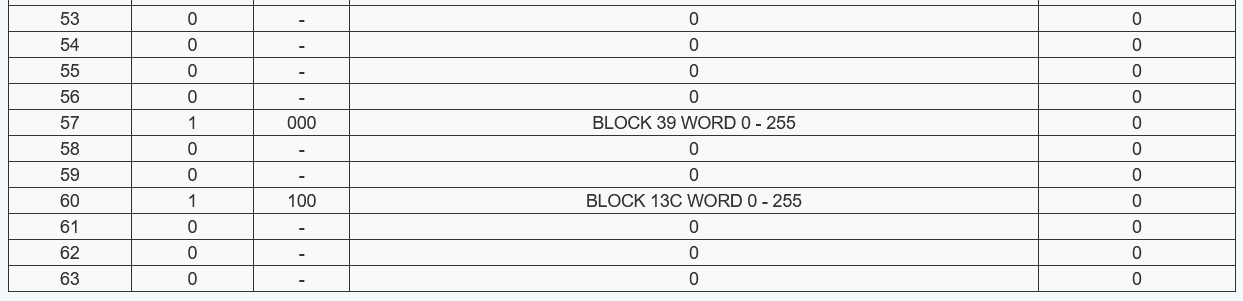
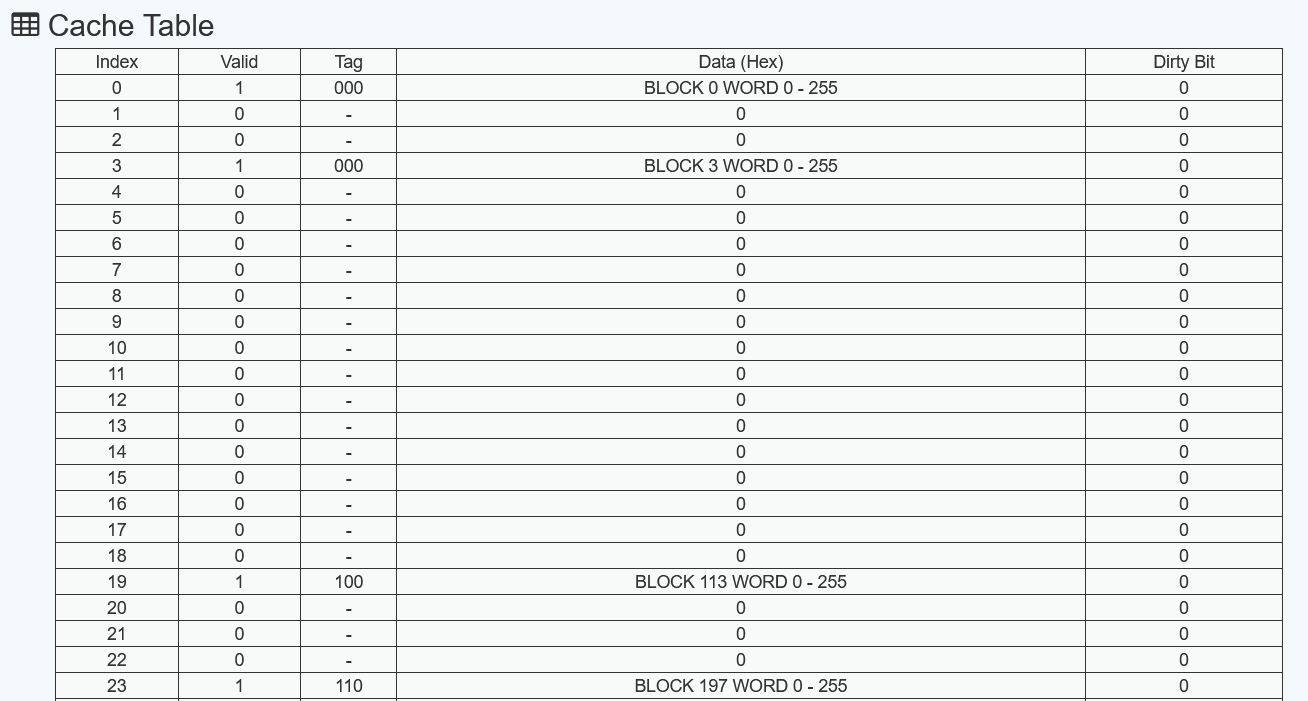
Week#\_\_\_\_8\_\_\_\_\_\_\_ Number: \_\_\_\_2\_\_

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.

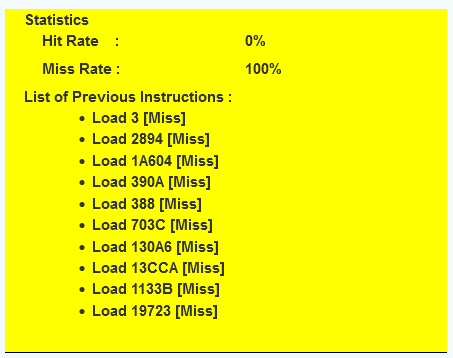
a) Cache Address Table showing the splitup of the address fields for the requests generated by the processor



b) Screenshot showing the Cache Table



c) Screenshot showing hit and miss rates



**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

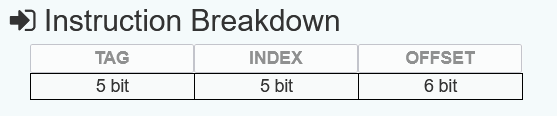
Date: 24/03/23

|  |  |  |
| --- | --- | --- |
| Name: Nikhil Girish | SRN: PES2UG21CS334 | Section: F |

Week#\_\_\_\_8\_\_\_\_\_\_\_ 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.

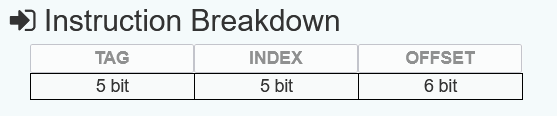
1. Calculate the number of bits in each of the Tag, Block, and Word fields of the memory address.



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

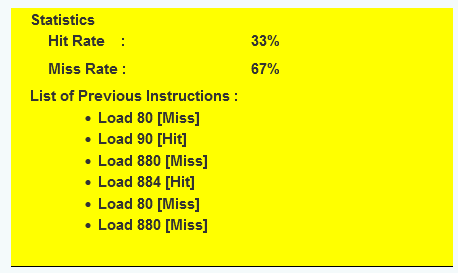
All the above addresses are shown in decimal values. Assume that the cache is initially empty. For each of the above addresses, indicate whether the cache access will result in a hit or a miss.

1. Cache Address Table showing the splitup of the address fields for the requests generated by the processor



1. Screenshot showing the Cache Table

c) Screenshot showing hit and miss rates



**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

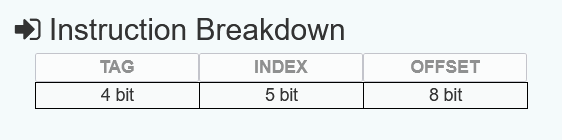
Date: 24/03/23

|  |  |  |
| --- | --- | --- |
| Name: Nikhil Girish | SRN: PES2UG21CS334 | Section: F |

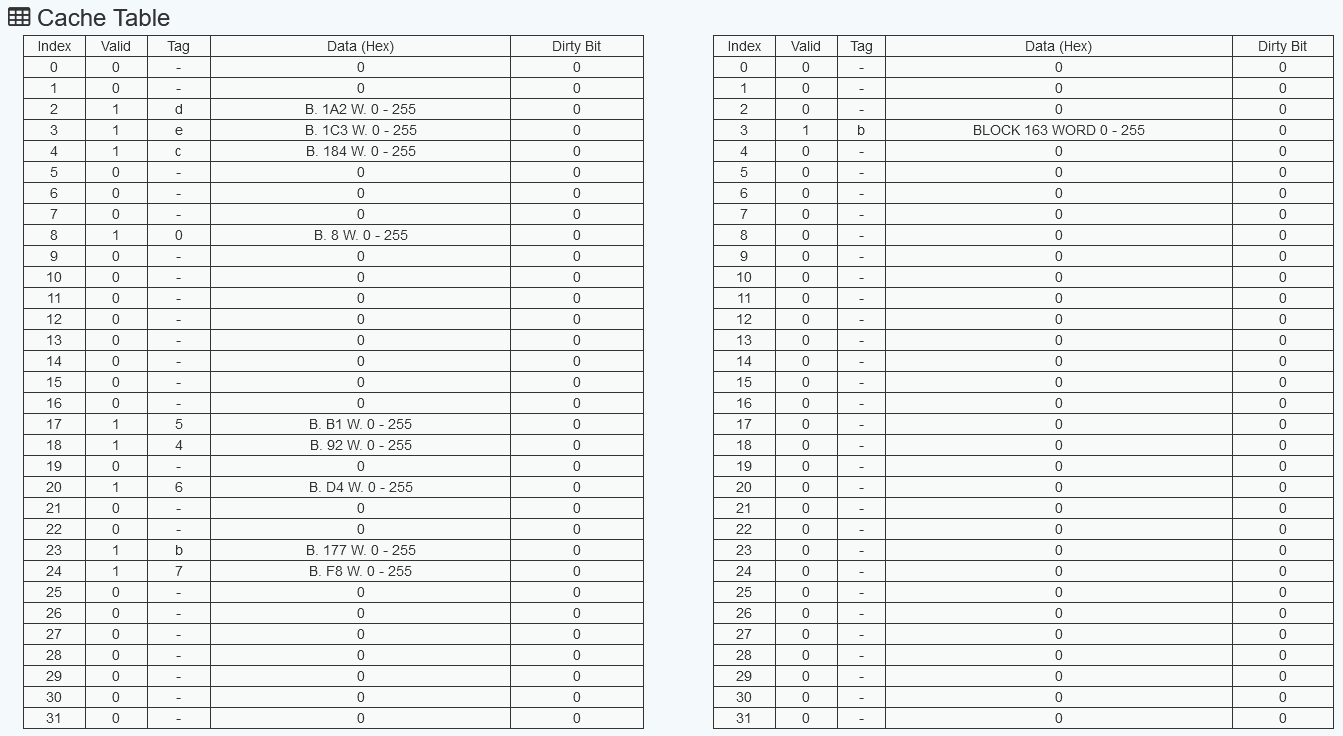
Week#\_\_\_\_8\_\_\_\_\_\_\_ Number: \_\_\_4\_\_

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.

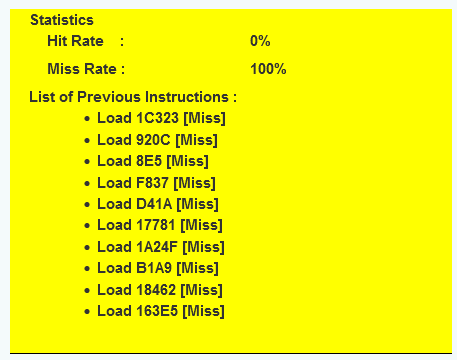
1. Cache Address Table showing the splitup of the address fields for the requests generated by the processor



1. Screenshot showing the Cache Table



1. Screenshot showing hit and miss rates



**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

Date: 24/03/23

|  |  |  |
| --- | --- | --- |
| Name: Nikhil Girish | SRN: PES2UG21CS334 | Section: F |

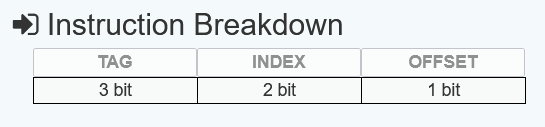
Week#\_\_\_\_8\_\_\_\_\_\_\_ Number: \_\_\_5\_\_

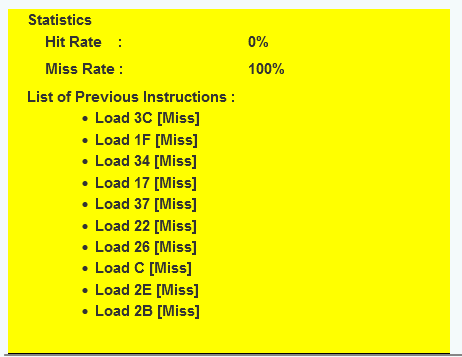
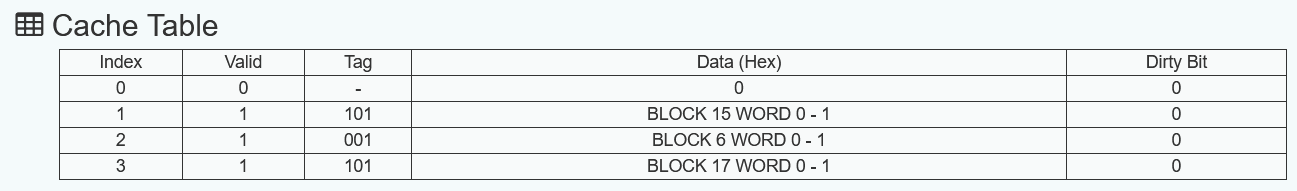
Consider a main memory having 64 byte capacity and cache memory of 8 bytes initially empty .The following addresses are generated by the CPU. All values in hexadecimal. Clearly label data that is replaced in cache lines

Show the cache memory table and filled data in the cache lines of block size 1 byte. LRU Policy is used.

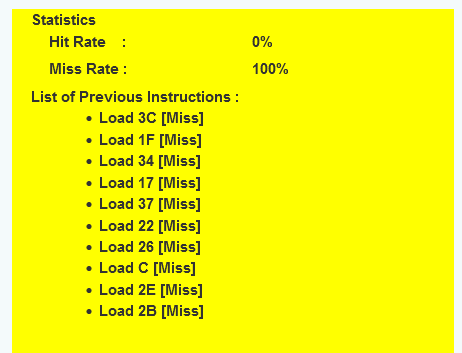
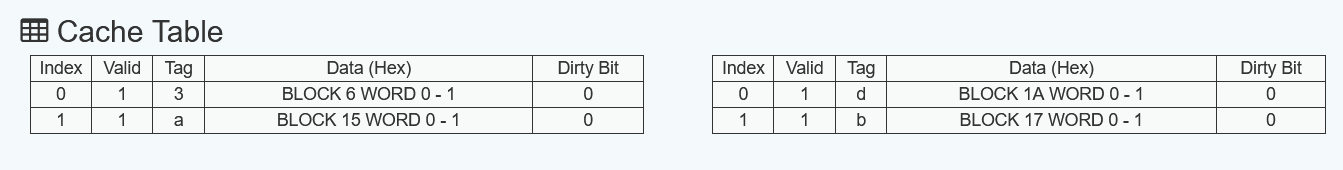
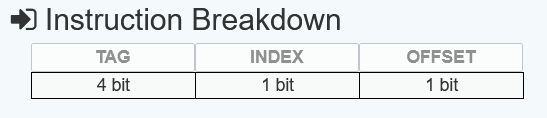
The cache is mapped as

1. Direct Mapped

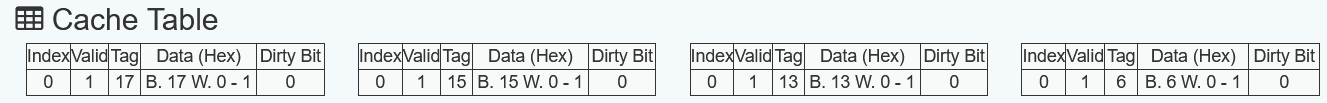
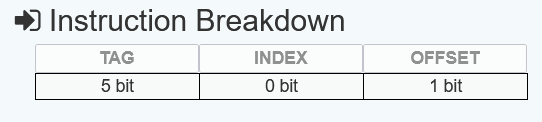


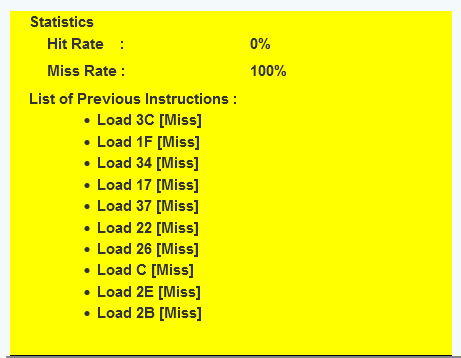


1. Two way set Associative

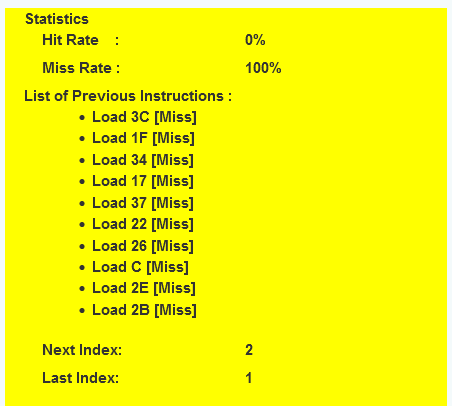
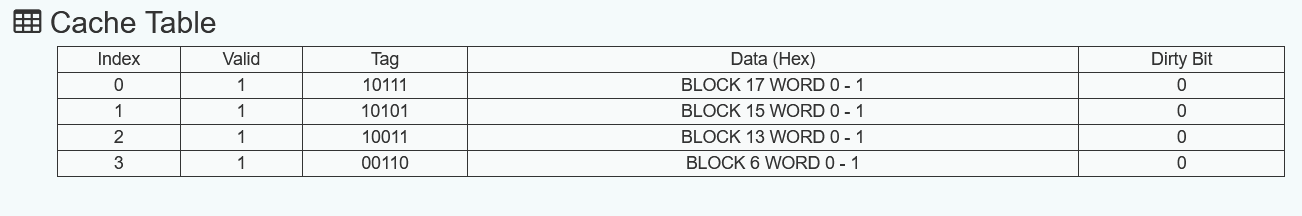
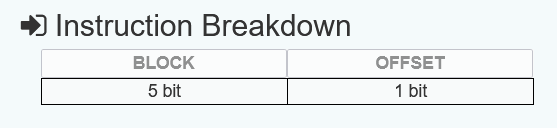


1. Four Way Set associative





1. Fully Associative



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

b) Screenshot showing the Cache Table

c) Screenshot showing hit and miss rates

**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: Nikhil Girish

Name: Nikhil

SRN: PES2UG21CS334

Section: 4F

Date: 24/03/23