

Documentation for Cache Project

Name : Yash Bhargava

Roll no : 2019289

Section : A

Group : 2

16 bit address input is assumed throughout the project.

Replacing, wherever required is done in FIFO manner.

Word size = 1 byte

Program CO1.java (Direct Mapping implementation)

Inputs :

Number of cache lines (power of 2; integer)

Block size (power of 2; integer)

Command (1 for write, 2 for read, 3 for display)

Address (16 bit string)

Data (integer)

Explanation : A 2-d array representing the cache is implemented. A 1-d array maintains the data of current blocks inside the cache. An index is fixed for a block which is derived by taking modulus of the address (sans word offset) with cl (cache lines).

Program CO2.java (Fully Associative implementation)

Inputs :

Number of cache lines (power of 2; integer)

Block size (power of 2; integer)

Command (1 for write, 2 for read, 3 for display)

Address (16 bit string)

Data (integer)

Explanation : A 2-d array representing the cache is implemented. A 1-d array maintains the data of current blocks inside the cache. A queue is maintained for implementation of replacing via FIFO method. A block can occupy any position in the cache.

Program CO3.java (n way Set Associative implementation)

Inputs :

Number of cache lines (power of 2; integer)

Block size (power of 2; integer)

N set (power of 2; integer)

Command (1 for write, 2 for read, 3 for display)

Address (16 bit string)

Data (integer)

Explanation : A 2-d array representing the cache is implemented. A 1-d array maintains the data of current blocks inside the cache. A queue is maintained for implementation of replacing via FIFO method. Blocks are clustered into small groups of size n, thereby dividing the cache into cl/n total groups.