**Experiment No.07**

Name : Sanskruti Shinde. (139)

**Aim :** Study of various cache mapping policies.

**Code :**

import java.util.Scanner;

public class CacheMapping {

    static int log2(int x) {

        return (int) (Math.log(x) / Math.log(2));

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Size of Cache memory (in KB): ");

        int cacheKB = sc.nextInt();

        int cacheSize = cacheKB \* 1024;

        System.out.print("Size of Main memory (in MB): ");

        int mainMB = sc.nextInt();

        int mainSize = mainMB \* 1024 \* 1024;

        int addrBits = log2(mainSize);

        System.out.println("Main memory address = " + addrBits + " bits");

        System.out.print("Size of each cache line (in Bytes): ");

        int lineSize = sc.nextInt();

        System.out.println("Select cache mapping policy: ");

        System.out.println("1. Direct mapping");

        System.out.println("2. 2-Way Set Associative mapping");

        System.out.println("3. Fully Associative mapping");

        int choice = sc.nextInt();

        int numCacheLines = cacheSize / lineSize;

        int numBlocks = mainSize / lineSize;

        System.out.println("\n=== Results ===");

        System.out.println("Size of Cache memory = " + cacheKB + " KB");

        System.out.println("Size of Main memory = " + mainMB + " MB");

        System.out.println("Main memory address = " + addrBits + " bits");

        System.out.println("Size of each cache line = " + lineSize + " Bytes");

        if (choice == 1) {

            System.out.println("Select cache mapping policy: Direct mapping");

            System.out.println("Number of cache banks = 1");

            System.out.println("Hence, size of cache bank = " + cacheKB + " KB");

            System.out.println("Cache lines per cache bank = " + cacheSize + "/" + lineSize +

                    " = " + numCacheLines + " (Line No-0 to Line No-" + (numCacheLines - 1) + ")");

            System.out.println("Number of main memory blocks = " + mainSize + "/" + lineSize +

                    " = " + numBlocks + " (Block -0 to Block No-" + (numBlocks - 1) + ")");

            int byteBits = log2(lineSize);

            int lineBits = log2(numCacheLines);

            int tagBits = addrBits - (lineBits + byteBits);

            System.out.println("Main memory address of " + addrBits + " bits is interpreted in 3 fields as calculated below:");

            System.out.println("LSB " + byteBits + " bits for Byte selection");

            System.out.println("Middle " + lineBits + " bits for Cache line selection");

            System.out.println("MSB " + tagBits + " bits (remaining) for the Tags");

            System.out.print("Input any Main memory block number for cache mapping = ");

            int blockNum = sc.nextInt();

            int lineNum = blockNum % numCacheLines;

            System.out.println("Block " + blockNum + " is mapped into cache line number = " + lineNum);

        } else if (choice == 2) {

            int sets = numCacheLines / 2;

            System.out.println("Select cache mapping policy: 2-Way Set Associative mapping");

            System.out.println("Number of sets = " + sets + " (Each set has 2 lines)");

            System.out.println("Cache lines per set = 2");

            System.out.println("Number of main memory blocks = " + numBlocks);

            int byteBits = log2(lineSize);

            int setBits = log2(sets);

            int tagBits = addrBits - (setBits + byteBits);

            System.out.println("Main memory address of " + addrBits + " bits is interpreted in 3 fields as:");

            System.out.println("LSB " + byteBits + " bits for Byte selection");

            System.out.println("Middle " + setBits + " bits for Set selection");

            System.out.println("MSB " + tagBits + " bits for Tags");

            System.out.print("Input any Main memory block number for cache mapping = ");

            int blockNum = sc.nextInt();

            int setNum = blockNum % sets;

            System.out.println("Block " + blockNum + " is mapped into Set number = " + setNum);

        } else {

          System.out.println("Select cache mapping policy: Fully Associative mapping");

            System.out.println("Number of cache lines = " + numCacheLines);

            System.out.println("Number of main memory blocks = " + numBlocks);

            int byteBits = log2(lineSize);

            int tagBits = addrBits - byteBits;

            System.out.println("Main memory address of " + addrBits + " bits is interpreted in 2 fields as:");

            System.out.println("LSB " + byteBits + " bits for Byte selection");

            System.out.println("MSB " + tagBits + " bits for Tags");

            System.out.print("Input any Main memory block number for cache mapping = ");

            int blockNum = sc.nextInt();

            System.out.println("Block " + blockNum + " can be mapped into ANY of the " + numCacheLines + " cache lines");

        }

        sc.close();

    }

}

**Output :**

