Laboratory Exercise 7
Cache Memory

Goals

After this laboratory exercise, you should understand the basic principles of cache memories and how different parameters of a cache memory affect the efficiency of a computer system.

Literature

- Patterson and Hennessy: Chapter 7.1–7.3
- MIPS Lab Environment Reference Manual

Preparation

Read the literature and this laboratory exercise in detail and solve the home assignments.

Home Assignment 1

The following C program contains two subroutines that return the sum of all the matrix elements. The only difference between the two subroutines is that they visit the matrix elements in a different order. This may seem unimportant, but with a cache memory, it may make a big difference.

```
/*
 *  Laboratory Exercise 7, Home Assignment 1
 *  Written by Jan Eric Larsson, 24 February 1999
 *
 */

#include <stdio.h>
#include <idt_entrypt.h>
#define N 10

int A[N][N];

int SumByColRow (int Matrix[N][N])
{
  int i, j, Sum = 0, Time;

  flush_cache();
  timer_start();
  for (j = 0; j < N; j ++) {
    for (i = 0; i < N; i ++) {
      Sum += Matrix[i][j];
    }
  }
  Time = timer_stop();
  printf("SumByColRow time: %d\n", Time);
  return Sum;
}

int SumByRowCol (int Matrix[N][N])
```

```
{
  int i, j, Sum = 0, Time;

  flush_cache();
  timer_start();
  for (i = 0; i < N; i ++) {
    for (j = 0; j < N; j ++) {
      Sum += Matrix[i][j];
    }
  }
  Time = timer_stop();
  printf("SumByRowCol time: %d\n", Time);
  return Sum;
}

main ()
{
  int a, b;

  printf ("Laboratory Exercise 8, Home Assignment 1\n");
  a = SumByColRow (A);
  b = SumByRowCol (A);
  printf ("The sums are %d and %d\n", a, b);
}
```

Study the code carefully so that you understand in what order the matrix elements are used. Also, explain in what order the matrix elements are located in physical memory.

Function of a Cache Memory

A cache memory is a memory that is smaller but faster than the main memory. Due to the locality of memory references, the use of a cache memory can have the effect on the computer system that the apparent speed of the memory is that of the cache memory, while the size is that of the main memory.

Assignment 1

Create a project, type in the C program of Home Assignment 1, build it and upload it to the cache simulator.

Assignment 2

Run the program in the cache simulator and study how the instruction cache works. Then give *full* answers to the following questions.

- How is the full 32-bit address used in the cache memory?
- What happens when there is a cache miss?
- What happens when there is a cache hit?
- What is the block size?
- What is the function of the tag?

Assignment 3

The parameters of the cache memory can be changed to test the effects of different cases. Investigate the effects of different parameter settings.

- Explain the following: cache size, block size, number of sets, write policy and replacement policy.
- If a cache is large enough that all the code within a loop fits in the cache, how many cache misses will there be during the execution of the loop? Is this good or bad?
- What should the code look like that would benefit the most from a large block size?

Cache Efficiency

The actual 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.

Assignment 4

Compile the C program of Home Assignment 1 with the compiler option *Optimization high*. Execute and look at the data cache. Use a cache size of 64, block size 16, and blocks in sets 1. Study the subroutines `SumByColRow` and `SumByRowCol`. Explain carefully in what order the memory addresses are visited by the two subroutines. Execute the program and study how many cache hits the two subroutines have. Is there a difference? Why?

Assignment 5

Download the summation program to the hardware and execute it with and without using a cache. In the `Project Settings` menu, under the `Link` command, you can set `Base Address`. If you set base address to `0x80020000`, the program will be started in cached instruction memory, while if you set base address to `0xA0020000`, it will be started in uncached instruction memory. What are the total execution times? Why?

Conclusions

Before you pass the laboratory exercise, think about the questions below:

- What is the general idea with cache memory?
- What is a block?
- How does block size affect the efficiency of a cache?
- How fast is a cache memory? How fast is a DRAM?
- Do the optimal cache parameters depend on the program code?
- How can one select good cache parameters?
- Is it possible to change cache size on a PC? On a Mac?