Laboratory Exercise 12

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

Preparation

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

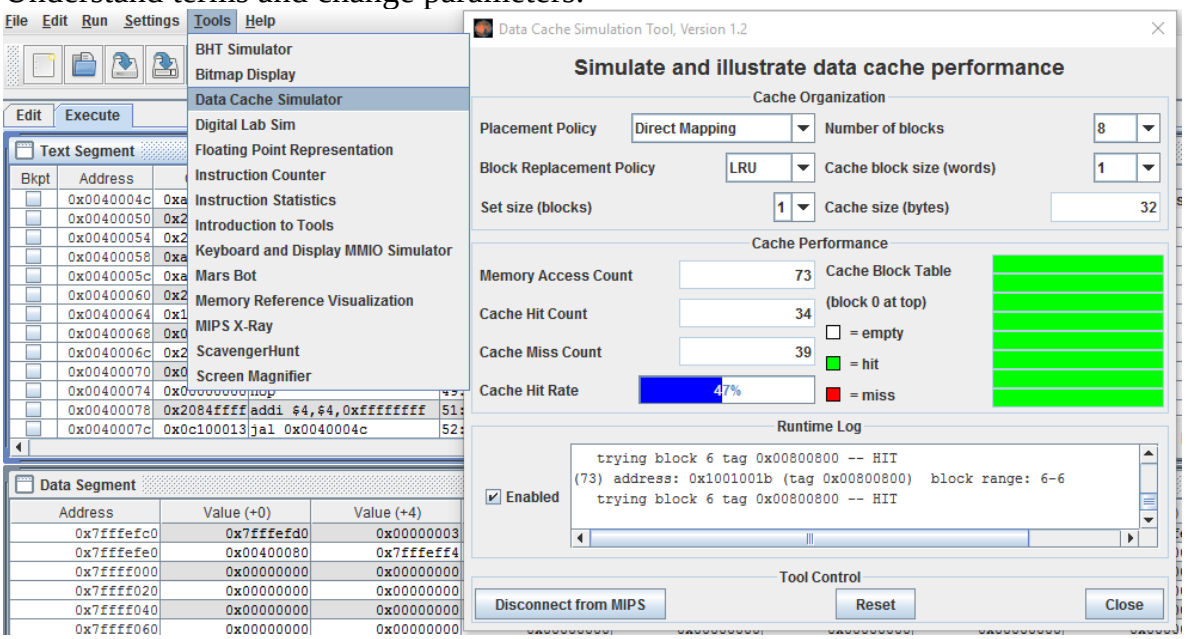
Assignments at Home and at Lab

Home Assignment 1

Looking for and research information about Cache in CPU.

Home Assignment 2

Try to use Data Cache Simulation Tool in MARS.
Understand terms and change parameters.



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 program of Laboratory 7, Home Assignment 4 , 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.

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?

Track Change

Ver	Date	Details
5.3	2013/04/23	More explanation for Lab Exercise 12, Home Assignment 2
6	2016/02	Using MARS
6.2	2018/02	Fix bug

Home Assignment 3

The following program sort a list of interger by ascending order. The swap procedure swap two successive elements. Read this example carefully, analyse each line of code, especially in sort procedure.

```
#Laboratory Exercise 5, Home Assignment 3
.data
test:       .word       3,6,2,1
.text
main:
      la    $a0,test
      li    $a1,4
      jal   sort
      nop
      la    $a0,test
      lw    $s0,0($a0)        #load test array after sorting
      lw    $s1,4($a0)
      lw    $s2,8($a0)
      lw    $s3,12($a0)
end_main:

swap:
      sll   $t1,$a1,2           #reg $t1=k*4
      add   $t1,$a0,$t1      #reg $t1=v+(k*4)
                                #reg $t1 has the address of v[k]
      lw    $t0,0($t1)          #reg $t0 (temp)=v[k]
      lw    $t2,4($t1)          #reg $t2=v[k+1]
                                #refers to next element of v
      sw    $t2,0($t1)          #v[k]=reg $t2
      sw    $t0,4($t1)          #v[k+1]=reg $t0 (temp)
      jr    $ra
end_swap:

#Procedure sort
#function: sort a list of element ascending
sort:
      addi  $sp,$sp,-20      #make room on stack for 5 registers
      sw    $ra,16($sp)      #save ra
      sw    $s3,12($sp)      #save $s3
      sw    $s2,8($sp)         #save $s2
      sw    $s1,4($sp)         #save $s1
      sw    $s0,0($sp)         #save $s0

      move  $s2,$a0          #copy parameter $$a0 into $s2 (save $a0)
      move  $s3,$a1          #copy parameter $a1 into $s3 (save a1)
      move  $s0,$zero          #i=0
for1tst:
      slt   $t0,$s0,$s3      #reg $t0=0 if $s0>=s3 (i>=n)
      beq   $t0,$zero,exit1    #go to exit1 if $s0>=s3 (i>=n)
      addi  $s1,$s0,-1         #j=i-1
for2tst:
      slti  $t0,$s1,0          #reg $t0=1 if $s1<0 (j<0)
      bne   $t0,$zero,exit2    #go to exit2 if $s1<0 (j<0)
      sll   $t1,$s1,2          #reg $t1=j*4
      add   $t2,$s2,$t1      #reg $t2=v+(j*4)
      lw    $t3,0($t2)         #reg $t3=v[j]
      lw    $t4,4($t2)         #reg $t4=v[j+1]
      slt   $t0,$t4,$t3      #reg $t0=0 if $t4>=t3
      beq   $t0,$zero,exit2    #go to exit2 if $t4>=t3
      move  $a0,$s2          #1st parameter of swap is v(old $a0)
      move  $a1,$s1          #2nd parameter of swap is j
```

```
      jal   swap
      addi  $s1,$s1,-1        #j=j-1
      j     for2tst           #jump to test of inner loop
exit2:
      addi  $s0,$s0,1         #i=i+1
      j     for1tst           #jump to test of outer loop
exit1:
      lw    $s0,0($sp)        #restore $s0
      lw    $s1,4($sp)        #restore $s1
      lw    $s2,8($sp)        #restore $s2
      lw    $s3,12($sp)     #restore $s3
      lw    $ra,16($sp)     #restore ra
      addi  $sp,$sp,20        #restore stack pointer
      jr    $ra
```

end_sort:

Assignment 4

Write a program that let user input a string (press Enter to terminate), sort this string by ascending order (based on ASCII code) and print the string after sorting.