

Laboratory Exercise 12 – Report:

Cache Memory

Lê Văn Duẩn - 20194508

1. Chạy code Lab7 Assignment 4

- Mã nguồn lab 7 Assignment 4:

```
assign4.asm
1  #Laboratory Exercise 7 Home Assignment 4
2
3  .data
4  Message: .asciiz "Ket qua tinh giai thua la: "
5  .text
6  main:   jal WARP
7  print:  add $a1, $v0, $zero # $a0 = result from N!
8          li $v0, 56
9          la $a0, Message
10         syscall
11  quit:   li $v0, 10 #terminate
12         syscall
13  endmain:
14  #-----
15  #Procedure WARP: assign value and call FACT
16  #-----
17  WARP:   sw $fp, -4($sp) #save frame pointer (1)
18         addi $fp, $sp, 0 #new frame pointer point to the top (2)
19         addi $sp, $sp, -8 #adjust stack pointer (3)
20         sw $ra, 0($sp) #save return address (4)
21         li $a0, 6 #load test input N
22         jal FACT #call fact procedure
23         nop
24         lw $ra, 0($sp) #restore return address (5)
25         addi $sp, $fp, 0 #return stack pointer (6)
26         lw $fp, -4($sp) #return frame pointer (7)
27         jr $ra
28  wrap_end:
29  #-----
30  #Procedure FACT: compute N!
31  #param[in] $a0 integer N
32  #return $v0 the Largest value
33  #-----
34  FACT:   sw $fp, -4($sp) #save frame pointer
```

```

35      addi $fp,$sp,0 #new frame pointer point to stack's top
36      addi $sp,$sp,-12 #allocate space for $fp,$ra,$a0 in stack
37      sw $ra,4($sp) #save return address
38      sw $a0,0($sp) #save $a0 register
39      slti $t0,$a0,2 #if input argument N < 2
40      beq $t0,$zero,recursive #if it is false ((a0 = N) >=2)
41      nop
42      li $v0,1 #return the result N!=1
43      j done
44      nop
45  recursive:
46      addi $a0,$a0,-1 #adjust input argument
47      jal FACT #recursive call
48      nop
49      lw $v1,0($sp) #load a0
50      mult $v1,$v0 #compute the result
51      mflo $v0
52  done:  lw $ra,4($sp) #restore return address
53         lw $a0,0($sp) #restore a0
54         addi $sp,$fp,0 #restore stack pointer
55         lw $fp,-4($sp) #restore frame pointer
56         jr $ra #jump to calling
57  fact_end:
58

```

Chạy và theo dõi trên Data Cache Simulator:

Data Cache Simulation Tool, Version 1.2

Simulate and illustrate data cache performance

Cache Organization

Placement Policy: **Direct Mapping** | Number of blocks: **8**

Block Replacement Policy: **LRU** | Cache block size (words): **4**

Set size (blocks): **1** | Cache size (bytes): **128**

Cache Performance

Memory Access Count: **22** | Cache Block Table (block 0 at top):

Cache Hit Count: **16** | ☐ = empty

Cache Miss Count: **6** | ☒ = hit

Cache Hit Rate: **73%** | ☐ = miss

Runtime Log

☐ Enabled

Tool Control

Disconnect from MIPS | **Reset** | **Close**

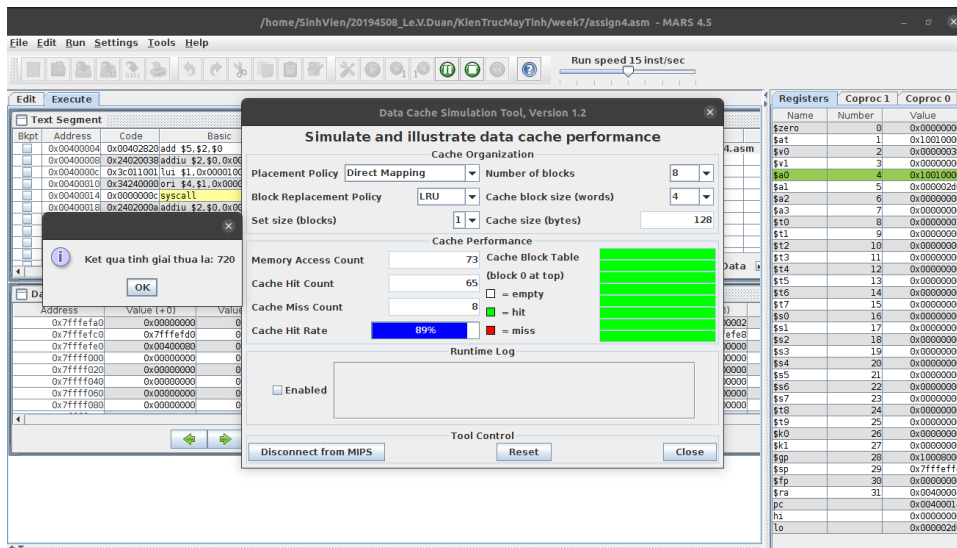
2. Cache Hit Count, Cache Miss count, ...

- Cache Hit count: số lần CPU yêu cầu truy cập vào Cache Memory thành công
- Cache Miss count: số lần CPU yêu cầu truy cập vào Cache Memory thất bại và phải truy cập vào bộ nhớ chính (Ram)

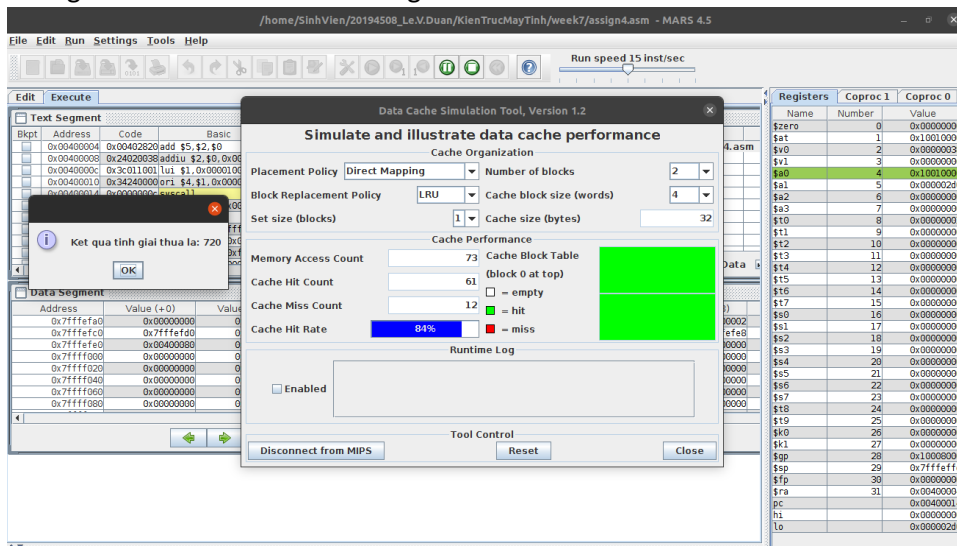
- Memory Access Count: số lần CPU yêu cầu truy cập vào Cache Memory (Tổng của Hit và Miss)
- Cache Hit Rate: tỷ lệ truy cập thành công vào Cache Memory (= Hit/Memory Access Count)
- Number of Blocks: số lượng block cho Cache Memory
- Cache block size (Words) : kích thước hay dung lượng của 1 block, như trong Mips mặc định words là 4 bytes
- Cache size (bytes): kích thước hay dung lượng của Cache Memory (= Number of blocks * Cache block size)
- Biểu diễn và mô tả hoạt động của Cache được hiển thị tại Cache Block table:

Cache Block Table	
(block 0 at top)	
<input type="checkbox"/> = empty	
<input checked="" type="checkbox"/> = hit	
<input type="checkbox"/> = miss	

- Khi Number of block tăng lên thì dẫn đến kích thước của Cache tăng lên -> Cache Hit count tăng lên và Cache Miss count giảm xuống.
- Ví dụ dưới đây thể hiện sự thay đổi khi chạy cùng 1 mã nguồn và thay đổi giá trị Number of block = 8



- Khi giảm Number of blocks xuống 2:



Cache Hit Rate giảm từ 89% xuống 84%

Cache Hit count: 65 -> 61

Cache Miss count: 8 -> 12