

Laboratory Exercise 12 – Report:

Cache Memory

Nguyễn Hải Dương- 20194530

1. Chạy code lab 7 Assignment 4

- Mã nguồn lab 7 Assignment 4:

```
1  #Laboratory Exercise 7, Home Assignment 4
2  .data
3  Message: .asciiz "Ket qua tinh giai thua la: "
4  .text
5  main: jal WARP
6
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 -
16 #Procedure WARP: assign value and call FACT
17 #-----
18 -
19 WARP: sw $fp,-4($sp) #save frame pointer (1)
20 addi $fp,$sp,0 #new frame pointer point to the top (2)
21 addi $sp,$sp,-8 #adjust stack pointer (3)
22 sw $ra,0($sp) #save return address (4)
23 li $a0,6 #load test input N
24 jal FACT #call fact procedure
25 nop
26
27 lw $ra,0($sp) #restore return address (5)
28 addi $sp,$fp,0 #return stack pointer (6)
29 lw $fp,-4($sp) #return frame pointer (7)
30 jr $ra
31 wrap_end:
32 #-----
33 -
34 #Procedure FACT: compute N!
```

```

34 #Procedure FACT: compute N!
35 #param[in] $a0 integer N
36 #return $v0 the largest value
37 #-----
38 -
39 FACT: sw $fp,-4($sp) #save frame pointer
40 addi $fp,$sp,0 #new frame pointer point to stack's
41 top
42 addi $sp,$sp,-12 #allocate space for $fp,$ra,$a0 in
43 stack
44 sw $ra,4($sp) #save return address
45 sw $a0,0($sp) #save $a0 register
46
47 slti $t0,$a0,2 #if input argument N < 2
48 beq $t0,$zero,recursive#if it is false ((a0 = N) >=2)
49 nop
50 li $v0,1 #return the result N!=1
51 j done
52 nop
53 recursive:
54 addi $a0,$a0,-1 #adjust input argument
55 jal FACT #recursive call
56 nop
57 lw $v1,0($sp) #load a0
58 mult $v1,$v0 #compute the result
59 mflo $v0
60 done: lw $ra,4($sp) #restore return address
61 lw $a0,0($sp) #restore a0
62 addi $sp,$fp,0 #restore stack pointer
63 lw $fp,-4($sp) #restore frame pointer
64 jr $ra #jump to calling
65 fact_end:
66

```

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

Cache Hit Count 16 (block 0 at top)

Cache Miss Count 6 ☐ = empty

Cache Hit Rate 73% ☒ = hit

☒ = miss

Runtime Log

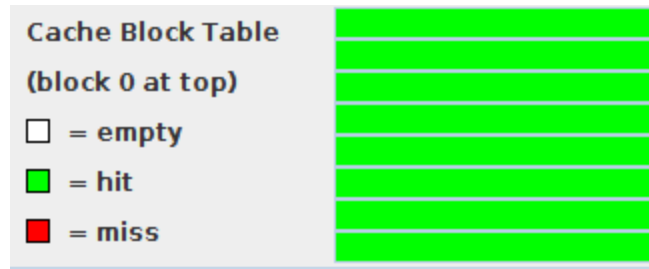
☐ Enabled

Tool Control

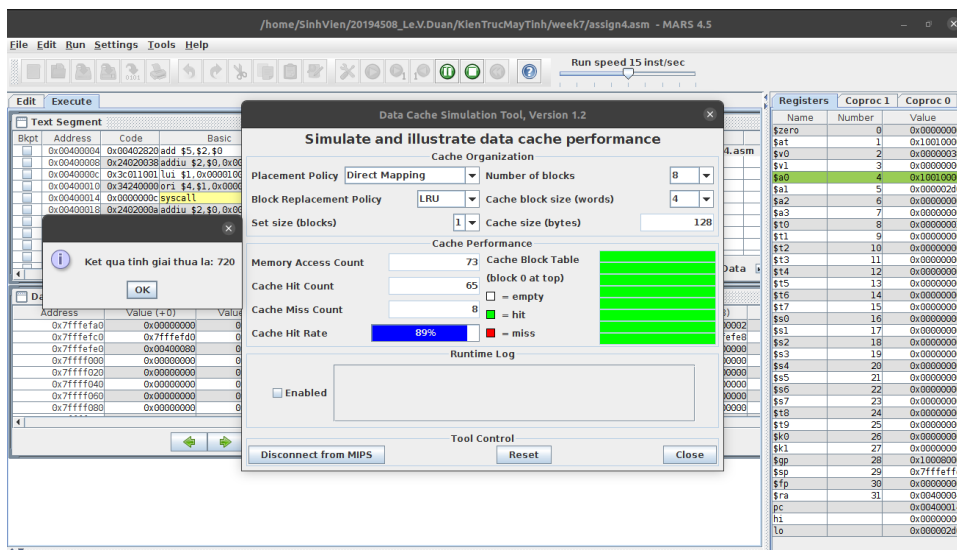
Disconnect from MIPS Reset Close

1. Cache Hit Count, Cache Miss count, ...

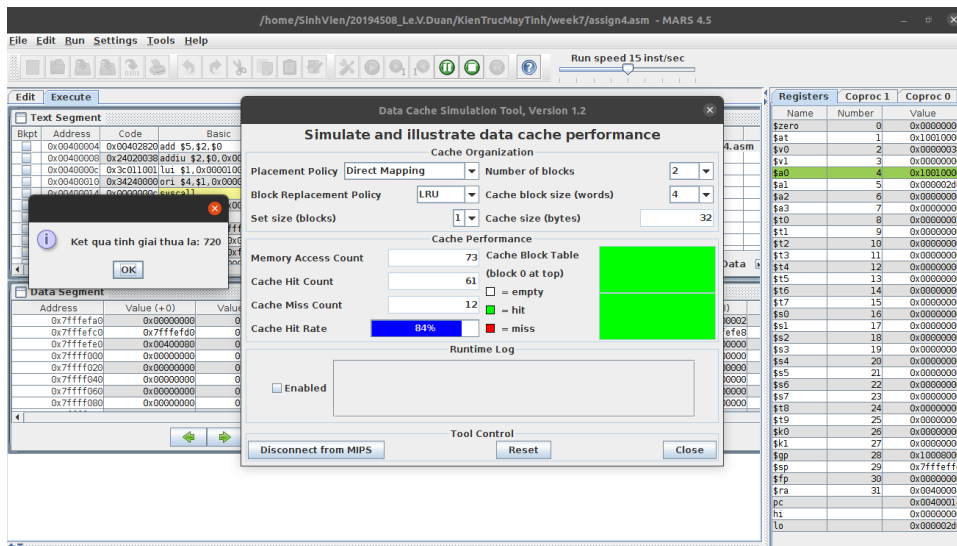
- Cache Hit count: số lần CPU yêu cầu truy cập vào Cache Memory thành công
- Cache Miss count: số lần CPU yêu cầu truy cập vào Cache Memory thất bại và phải truy cập vào bộ nhớ chính (Ram)
- Memory Access Count: số lần CPU yêu cầu truy cập vào Cache Memory (Tổng của Hit và Miss)
- Cache Hit Rate: tỷ lệ truy cập thành công vào Cache Memory ($= \text{Hit} / \text{Memory Access Count}$)
- 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 ($= \text{Number of blocks} * \text{Cache block size}$)
- Biểu diễn và mô tả hoạt động của Cache được hiển thị tại Cache Block table:



- 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