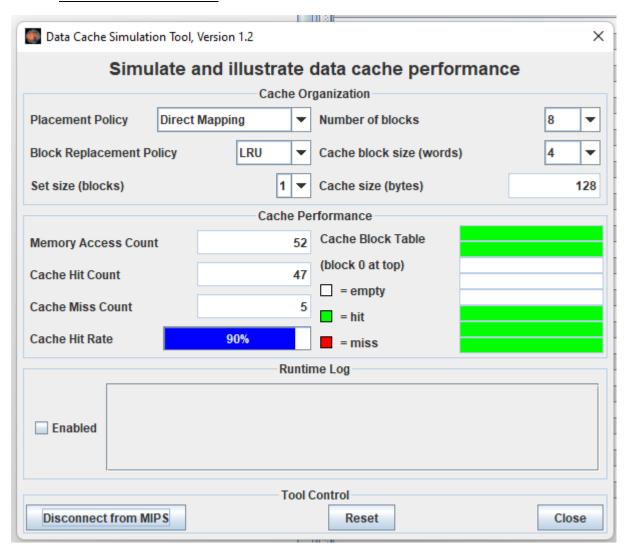
# **BÀI TẬP TUẦN 12**

## **Assignment 1:**

Chương trình:

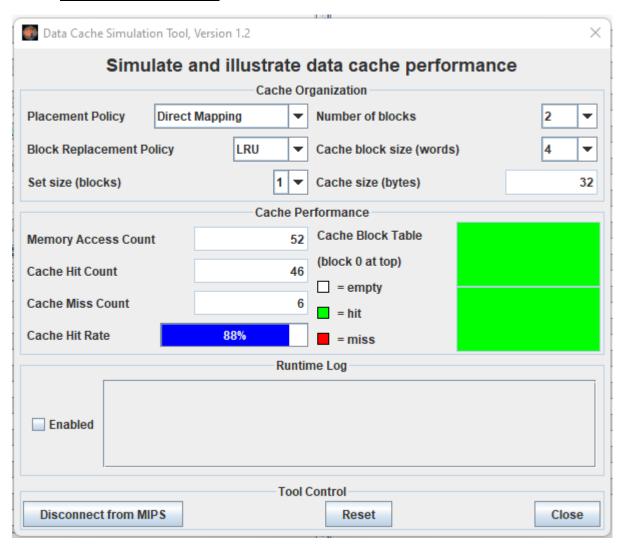
```
#Laboratory Exercise 7, Home Assignment 4
.data
Message:
              .asciiz "Ket qua tinh giai thua la: "
.text
              WARP
main: jal
              $al,$v0,$0,
print: add
                              #$a0 = result from N!
              $v0,56
       1 i
       la
               $a0,Message
       syscall
quit: li
               $v0,10
                              #ket thuc thuc hien
       syscall
endmain:
WARP: SW
               $fp,-4($sp)
                           #save frame pointer
       addi
              $fp,$sp,0
                             #new frame pointer point to the top
       addi
               $sp,$sp,-8
                              #adjust stack pointer
       SW
               $ra,0($sp)
                             #save return address
       1i
               $a0,3
                              #load test input N
               FACT
                              #call fact procedure
       jal
       nop
                            #restore return address
              $ra,0($sp)
       1 w
       addi
              $sp,$fp,0
                            #return stack pointer
       lw
               $fp,-4($sp)
                           #return frame pointer
       jr
wrap_end:
FACT:
               $fp,-4($sp)
                                     #save frame pointer
      SW
       addi
               $fp,$sp,0
                                     #new frame pointer point to stack's top
       addi
               $sp,$sp,-12
                                    #allocate space for $fp,$ra,$a0 in stack
       sw
               $ra,4($sp)
                                     #save return address
               $a0,0($sp)
                                     #save $a0 register
       sw
       slti
                                     #if input argument N < 2
              $t0,$a0,2
       beq
               $t0,$zero,recursive
                                   #if it is false ((a0 = N) >= 2)
       nop
               $v0,1
                                     #return the result N/=1
       1i
               done
       j
       nop
                    $a0,$a0,-1
recursive:
               addi
                                     #adjust input argument
                     FACT
               jal
                                     #recursive call
               nop
                     $v1,0($sp)
                                     #load a0
               mult $v1,$v0
                                     #compute the result
               mflo $v0
done:
               lw
                     $ra,4($sp)
                                     #restore return address
               1 w
                     $a0,0($sp)
                                     #restore a0
               addi $sp,$fp,0
                                     #restore stack pointer
                     $fp,-4($sp)
                                     #restore frame pointer
              1 w
               jr
                     $ra
                                     #jump to calling
fact_end:
```

- Giải thích:
  - + Khi Number of blocks = 8



+ Khi Number of blocks tăng lên thì dẫn đến kích thước của Cache tăng lên -> Cache Hit Count tăng lên và Cache Miss Count giảm xuống

### + Khi Number of blocks = 2



- + Khi Number of blocks giảm xuống thì dẫn đến kích thước của Cache giảm xuống -> Cache Hit Count giảm xuống và Cache Miss Count tăng lên
  - Cache Hit Rate giảm từ 90% xuống 88%
  - Cache Hit Count giảm từ 47 xuống 46
  - Cache Miss Count tăng từ 5 lên 6

#### **Assignment 2:**

Run the program in the cache simulator and study how the instruction cache works.

Then give full answers to the following questions.

- a, How is the full 32-bit address used in the cache memory?
- Lấy địa chỉ 32 bit, dịch trái 2 bit, và lưu vào phần tag của block cache.
- b, What happens when there is a cache miss?
- Khi bộ xử lý không tìm thấy vị trí bộ nhớ, trong bộ nhớ cache thì bộ nhớ cache đã miss.
- Khi lỗi bộ nhớ cache, bộ đệm sẽ phân bổ một mục nhập mới rồi sao chép dữ liệu từ bộ nhớ chính, sau đó yêu cầu được thực hiện từ nội dung của bộ nhớ cache.
- c, What happens when there is a cache hit?
- Khi bộ xử lý nhận thấy rằng vị trí bộ nhớ nằm trong bộ nhớ cache, thì dữ liệu sẽ được đọc từ bộ nhớ cache
- d. What is the block size?
- Bộ nhớ cache được chia làm nhiều block. Block size là kích thước của 1 block trên cache.
- e, What is the function of the tag?
- Tag là phần để lưu trữ địa chỉ của dữ liệu.

#### **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.

- a, Explain the following: cache size, block size, number of sets, write policy and replacement policy?
- Kích thước của bộ nhớ đệm (cache size) là lượng dữ liệu bộ nhớ chính mà nó có thể chứa. Kích thước này có thể được tính bằng số byte được lưu trữ trong mỗi khối dữ liệu nhân với số khối được lưu trữ trong bộ nhớ đệm.
- Block size là kích thước các khối chia ra từ bộ nhớ đệm. VD: Nếu tăng kích thước khối trong khi vẫn giữ nguyên kích thước bộ nhớ cache, thì sẽ giảm số khối mà bộ nhớ đệm có thể chứa.
- Bộ nhớ đệm được chia thành các nhóm khối (block), được gọi là sets.
- Write policy and replacement policy: Bộ đệm cải thiện hiệu suất bằng cách giữ các mục dữ liệu gần đây hoặc thường sử dụng ở các vị trí bộ nhớ truy cập nhanh hơn hoặc rẻ hơn về mặt tính toán so với các bộ nhớ lưu trữ thông thường. Khi bộ nhớ đệm đầy, thuật toán phải chọn mục

nào cần loại bỏ để nhường chỗ cho các mục mới.

b, 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?

- Số lần miss = Số lần đọc và ghi vào 1 địa chỉ mới.
- Nếu như số lần đọc, ghi vào cùng 1 địa chỉ nhiều thì đây là điều tốt.
- c, What should the code look like that would benefit the most from a large block size?
- Với blocksize lớn hơn, ta có thể lưu nhiều hơn trên 1 block nhưng số block bị giảm đi.