Assignment 6

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

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

1. How is the 32-bit address used in the cache memory?

- Mỗi 1 byte trong main memory sẽ cần 32 bits address để biểu diễn, 32 address liên tiếp nhau như thế sẽ tạo thành 32 bytes blocks và được biểu diễn trong cache
- Mỗi cache block chứa 32 bytes, **trường offset (offset field)** chứa 5 bits (2^5 = 32) - từ 0 đến 4, tiếp theo để có thể đánh index các block của cache, tiếp tục sử dụng 5 bit nữa: từ 5 đến 9 để biểu diễn **trường index, hoặc set (set field)** . Trường còn lại là **trường tag (tag field),** chiếm các bits còn lại.

2. What happens when there is a cache miss?

- Khi cache miss diễn ra, hệ thống sẽ tiếp tục tìm dữ liệu ở trong main memory, sau đó lưu dữ liệu đó vào bộ nhớ đệm cache, việc lưu như thế nào còn phụ thuộc vào replacement policy
- Tỉ lệ cache hit rate giảm khi cache miss diễn ra

3. What happens when there is a cache hit?

- Khi mà hệ thống yêu cầu tìm dữ liệu, thì CPU sẽ đi vào bộ nhớ cache để tìm, nếu thấy dữ liệu đó được lưu trong bộ nhớ cache thì được gọi là cache hit.
- Tỉ lệ cache hit rate tăng khi cache hit diễn ra

4. What is the block size?

- Block size là một khối, chứa 32 địa chỉ, mỗi địa chỉ được biểu diễn bởi 32 bit, được hiểu là 1 byte.
- Vì thế nên block size ở đây là 32 byte blocks,là kích cỡ của một khối được lưu các đia chỉ liên tiếp.

5. What is the function of the tag?

- Thứ nhất, cần hiểu rằng volume của cache nhỏ hơn main memory rất nhiều, vì thế nên các 32 bytes blocks ở main memory sẽ sử dụng chung 32 bytes blocks ở cache
- Trường offset chỉ có chức năng chỉ rõ là 32 bits address của trong một khối 32 bytes blocks, chứ không phân biệt được 32 bytes blocks với nhau
- Chính vì thế nên cần trường ttag (tag-field) → tag-field chỉ rõ ra là khối blocks 32 bytes nào của main memory khi sử dụng chung 32 bytes block trong cache

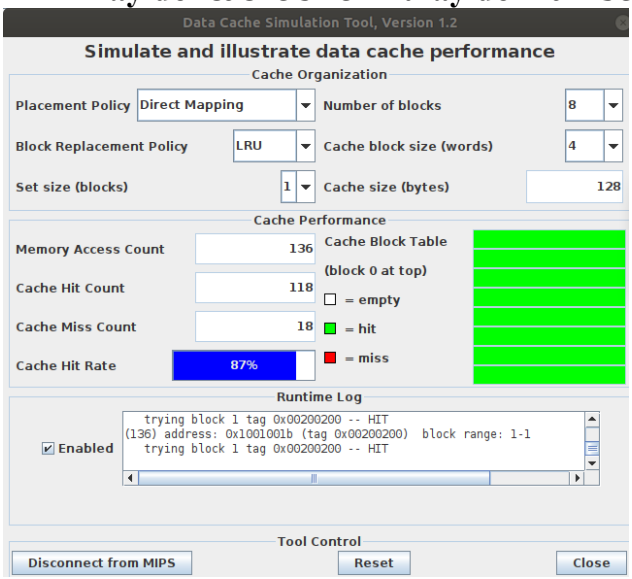
Assignment 7

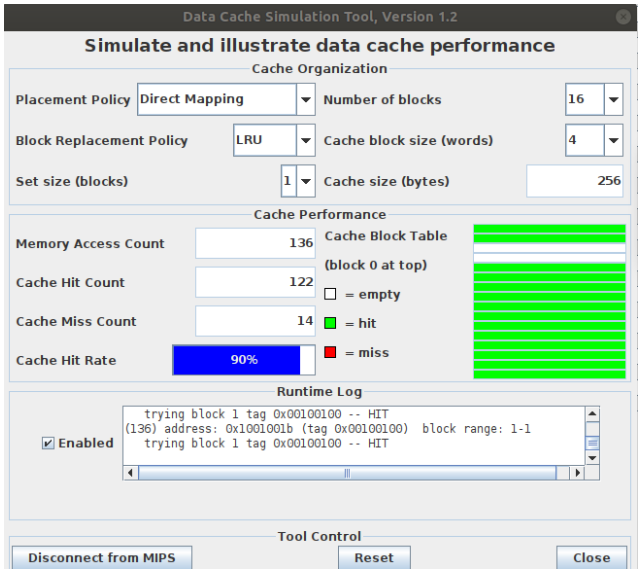
The parameters of the cache memory can be changed to test the effects of different cases. Investigate the effects of different parameter settings.

1. Explain the following: cache size, block size, number of sets, write policy and replacement policy.
2. 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?
3. What should the code look like that would benefit the most from a large block size?

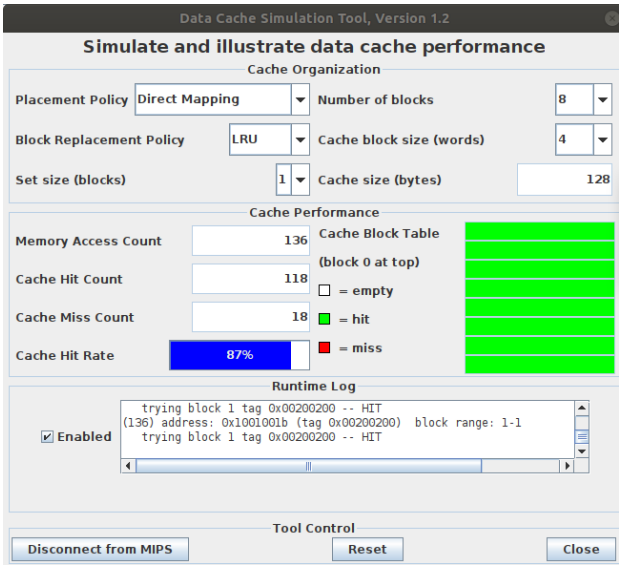
❖ Hiện tượng quan sát:

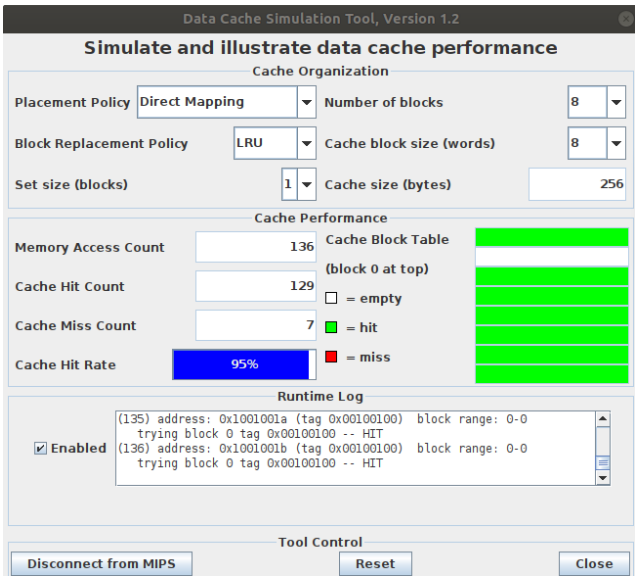
- **Thay đổi cache size -< thay đổi number of blocks**





- **Thay đổi block size → cache size bị thay đổi theo:**

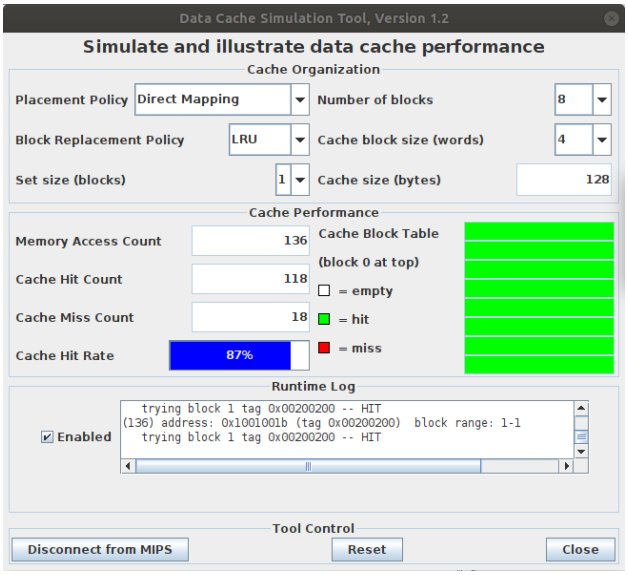


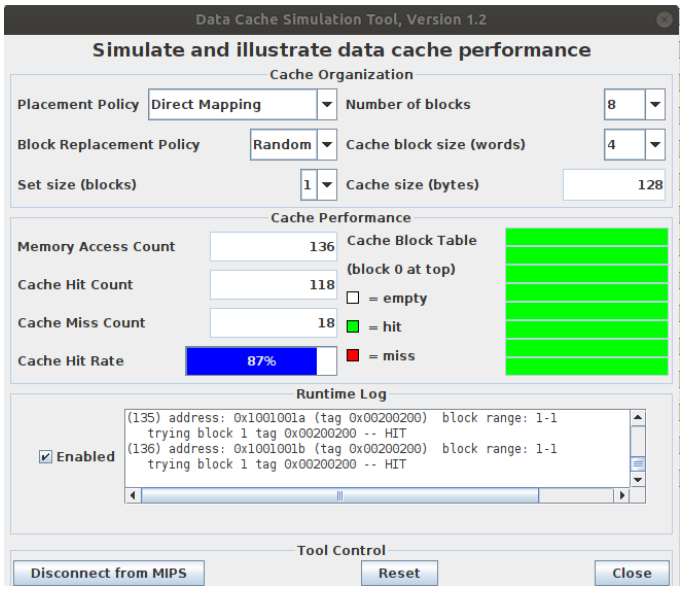


- **Thay đổi number of sets (number of block)** → tương tự như thay đổi cache size

- **Thay đổi write policy** → Kiến trúc mips không hỗ trợ thay đổi write policy trong data cache simulator tool, nhiều khả năng được mặc định là write - through policy

- **Thay đổi replacement policy**

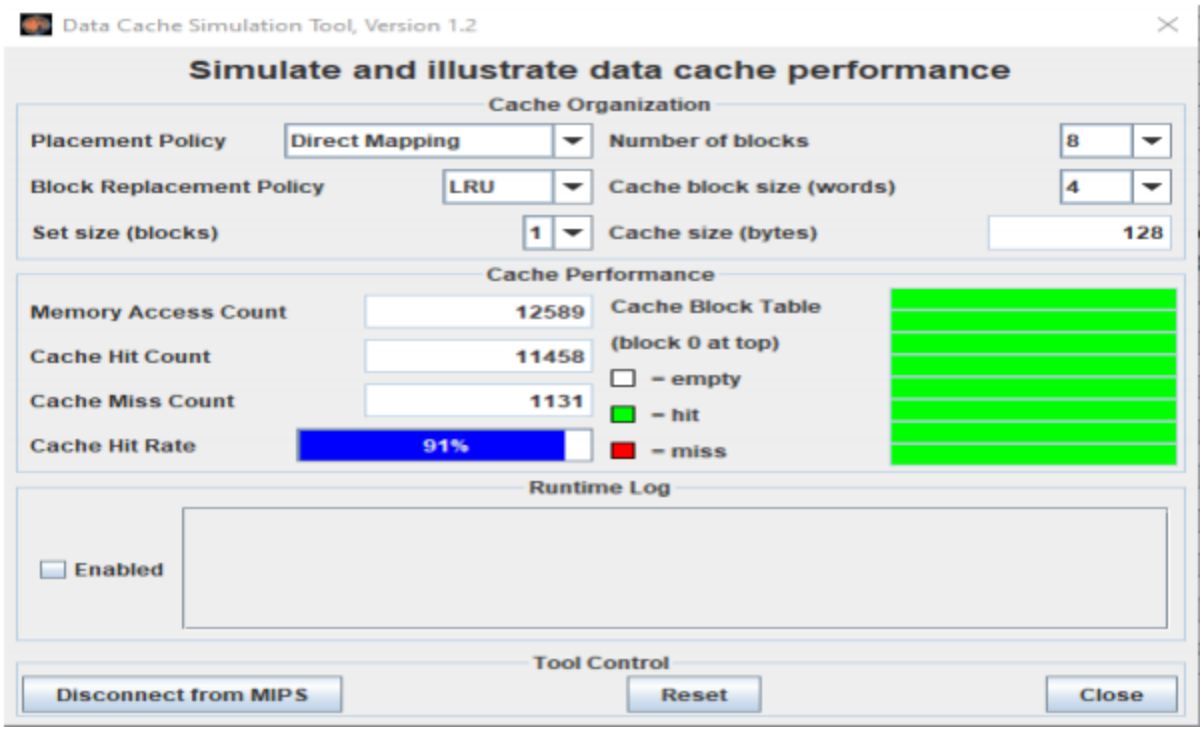




❖ Nhận xét - giải thích

- **Cache size:**
  - Nếu cache size quá lớn sẽ ảnh hưởng bất lợi đến độ trễ của cache hit và cache miss → càng lớn thì càng chậm
  - Nếu cache size nhỏ quá thì không thể tận dụng được cơ chế hoạt động của bộ nhớ cache - temporal locality, những dữ liệu cũ sẽ bị thay thế liên tục
  - Temporal locality : mỗi lần truy xuất 1 phần tử nào trong bộ nhớ thì sẽ mang phần tử này vào trong bộ nhớ cache. Với hy vọng phần tử này sẽ được truy xuất lại, khi được truy xuất lại sẽ được truy xuất trong bộ nhớ cache, làm giảm thời gian truy xuất.
  - Trong trường hợp trên , việc tăng cache size làm tăng cache hit rate
- **Block size:**
  - Nếu block size quá nhỏ thì sẽ không tận dụng tối ưu được spartial locality
    Spatial locality: Mỗi lần truy xuất 1 byte thì sẽ mang luôn những byte lân cận vào trong bộ nhớ cache với hy vọng những byte lân cận sẽ được truy xuất tiếp theo.
  - Nếu block size quá lớn, thì tổng số blocks sẽ quá ít, những dữ liệu có ích sẽ thường xuyên bị thay thế, và cần nhiều thời gian hơn để điền vào bộ nhớ đệm cache. Ngoài ra cache blocks quá lớn có thể lãng phí “bus bandwidth”, nghĩa là bus bandwith chia block ra thành các block nhỏ, và liên kết từng subblock bằng các “valid bits”
  - Trong trường hợp trên, thay đổi block size - tăng lên, làm tăng cache hit rate
- **Replacement policy:**
  - Cả LRU và Random đều cho average hit rate giống nhau
  - Điểm khác biệt là Random cho performance của cache với high associativity.
  - Trường hợp trên, cả Random và LRU đều cho cache hit rate giống nhau

1. Explain the following: cache size, block size, number of sets, write policy and replacement policy.
    - Cache size, block size
        - Kích thước bộ nhớ cache hoặc kích thước khối tăng lên làm cho tỷ lệ bộ nhớ cache bị bỏ sót giảm xuống, bởi vì số lượng khối có thể được giữ trong bộ nhớ cache sẽ trở nên nhỏ và sẽ có rất nhiều cạnh tranh cho các khối đó.
        - Khi tăng số lượng khối và kích thước khối bộ nhớ cache, tỷ lệ truy cập bộ nhớ cache tăng đột ngột, bộ đếm số lần bỏ lỡ bộ nhớ cache cũng giảm đi rất nhiều.
    - Number of sets
        - Số lượng bộ càng nhiều thì chương trình càng ít bỏ lỡ bộ nhớ cache, vì số bộ phụ thuộc vào số khối và kích thước bộ nhớ cache.
        - Tập hợp chứa một khối bộ nhớ được cho bởi (Số khối) modulo (Số bộ trong bộ nhớ cache).
    - Write policy, replacement policy
        - Nếu chúng ta sử dụng ghi thông qua, nó sẽ yêu cầu ghi nhiều bộ nhớ chính hơn vì mỗi lần dữ liệu được ghi vào bộ nhớ cache, chúng ta cũng ghi dữ liệu vào bộ nhớ chính.
        - Nếu chúng ta sử dụng ghi ngược (một bit bẩn (D) được liên kết với mỗi khối bộ nhớ cache. D là 1 khi khối bộ nhớ cache đã được ghi và 0 nếu không.
        - Các khối bộ nhớ cache bẩn chỉ được ghi trở lại bộ nhớ chính khi chúng bị loại bỏ khỏi bộ nhớ đệm) nó sẽ yêu cầu ghi ít hơn vào bộ nhớ chính so với ghi qua, khi đó chương trình sẽ có hiệu suất bộ nhớ cache tốt hơn.
    - Replacement policy
        - Chính sách LRU làm tăng bộ nhớ cache. Ít hơn với chính sách ngẫu nhiên.
        - Việc thay thế LRU được thực hiện bằng cách theo dõi thời điểm từng phần tử trong tập hợp được sử dụng so với các phần tử khác trong tập hợp.
        - Đối với bộ đệm ẩn kết hợp bộ hai chiều, việc theo dõi thời điểm hai phần tử được sử dụng có thể được triển khai bằng cách giữ một bit duy nhất trong mỗi bộ và đặt bit để chỉ ra một phần tử bất cứ khi nào phần tử đó được tham chiếu.
        - Khi tính liên kết tăng lên, việc triển khai LRU trở nên khó khăn hơn
2. If a cache memory is so large that all the code of a loop fits into the cache memory, how many cache misses will occur during the execution of the loop? Is this good or bad?
    - Khi bộ nhớ đệm quá lớn sao cho tất cả code của một vòng lặp vừa với bộ nhớ đệm, thì số lượng cache misses sẽ bằng với số lượng lệnh phải truy cập vào bộ nhớ cache ở vòng lặp đầu tiên. Tới các vòng lặp tiếp theo, số lượng cache misses = 0.
    - Tuy nhiên, nếu bộ nhớ đệm quá lớn thì dựa theo số lần truy cập, thì thời gian sẽ tăng lên → slower access, ví dụ phải truy cập nhiều hơn vào bộ nhớ đệm để vận chuyển những “cache miss” - những dữ liệu chưa có.
    - Nhìn chung, phụ thuộc vào hệ thống chương trình, có lẽ cache càng lớn thì càng hữu ích hơn, vì các chương trình trên máy tính bây giờ yêu cầu nhiều dữ liệu
3. What should a code (i.e. program) look like to get the most benefit of a large block size?
    - Số lượng dữ liệu/ địa chỉ yêu cầu tìm nên nằm trong khoảng block size
    - Kích thước khối lớn giúp giảm tổng số khối trong bộ nhớ cache, nhưng nó cũng có thể làm tăng sự thiếu hụt xung đột, vì nhiều địa chỉ hơn sẽ ánh xạ vào cùng một tập hợp và có thể gây ra xung đột. Để có được nhiều lợi ích nhất của kích thước khối lớn, mã phải chứa các mảng, ngăn xếp (cấu trúc dữ liệu tuần tự). Trong Mã mẫu 2 là một ví dụ về việc tận dụng lợi thế của kích thước khối lớn.



- Theo mặc định, tỷ lệ truy cập bộ nhớ cache ở mức 91%, có hơn 1000 lần bộ nhớ cache đã xảy ra. Tuy nhiên, khi tăng kích thước khối cache thì Cache Hit Rate tăng đột ngột và việc bỏ sót cache gần như không xảy ra.