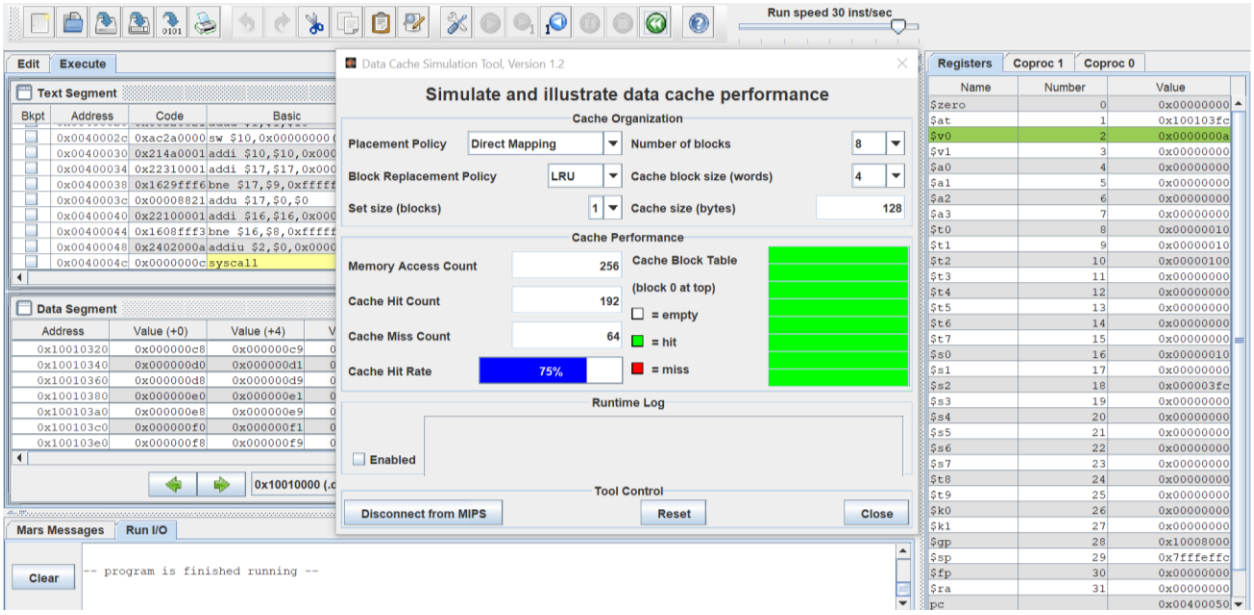
Báo cáo Thực hành KTMT buổi 13

Họ và tên: Nguyễn Đức Phú

MSSV: 20215116

Part 1: Running the Data Cache Simulator tool

8. What was the final cache hit rate? => 75%



9. Thay đổi block size và dự đoán kết quả

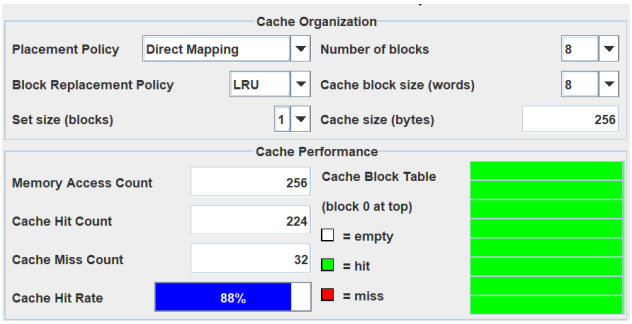
- Tăng từ 4 lên 8: Với block size là 8 → mỗi lần cache miss thì 7 phần tử sau đó sẽ được tìm thấy tại cache

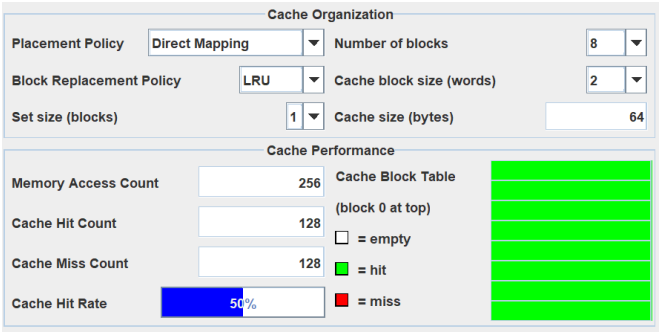
⇨ Tỷ lệ dự đoán: 7/8 (= 87.5%)

- Giảm từ 4 xuống 2: Với block size là 2 → mỗi lần cache miss thì 1 phần tử sau đó sẽ được tìm thấy tại cache

⇨ Tỷ lệ dự đoán: ½ (= 50%)

10. Chạy và kiểm chứng

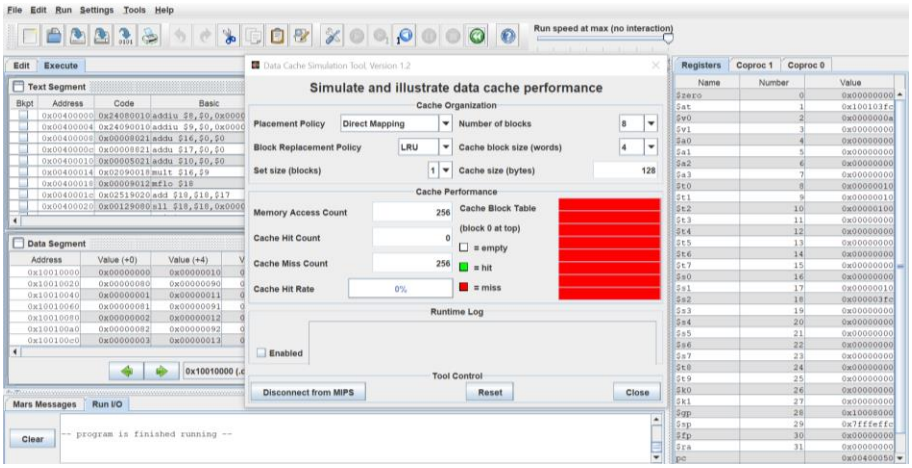




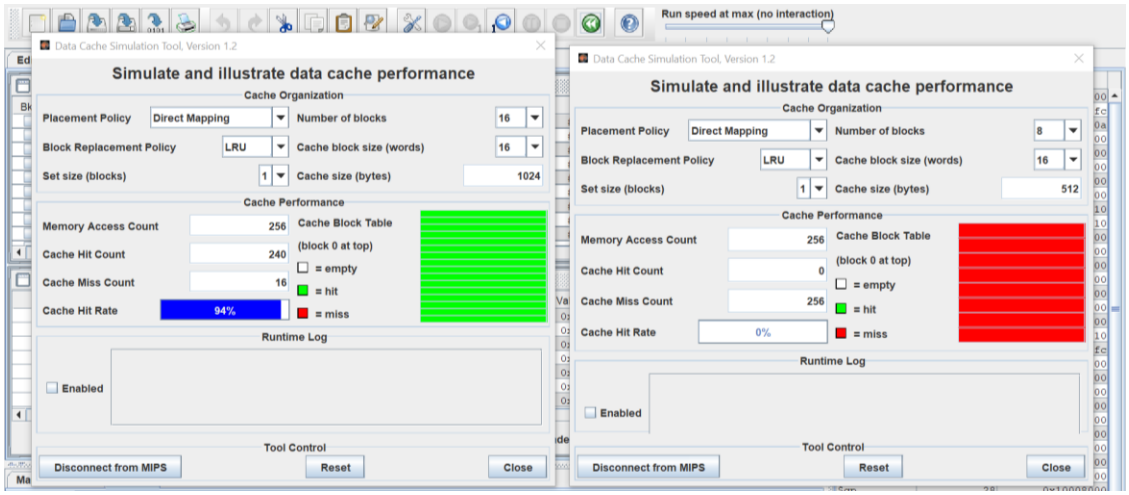
Con số 88% là do làm tròn từ 87.5%

12. Chạy với file **`column-major.asm`**

What was the cache performance for this program? => 0%



14. Chạy thử song song



15. Nhận xét

- Nếu chỉ tăng block size lên 16 sẽ không giúp cải thiện hiệu suất do vẫn chưa phù hợp với kích thước ma trận

⇨ Các block ban đầu bị thay thế trước khi chúng được truy cập ở phần sau của chương trình

⇨ Cache hit rate: 0%

- Nếu tăng số block lên 16 kích thước ma trận sẽ vừa với bộ đệm

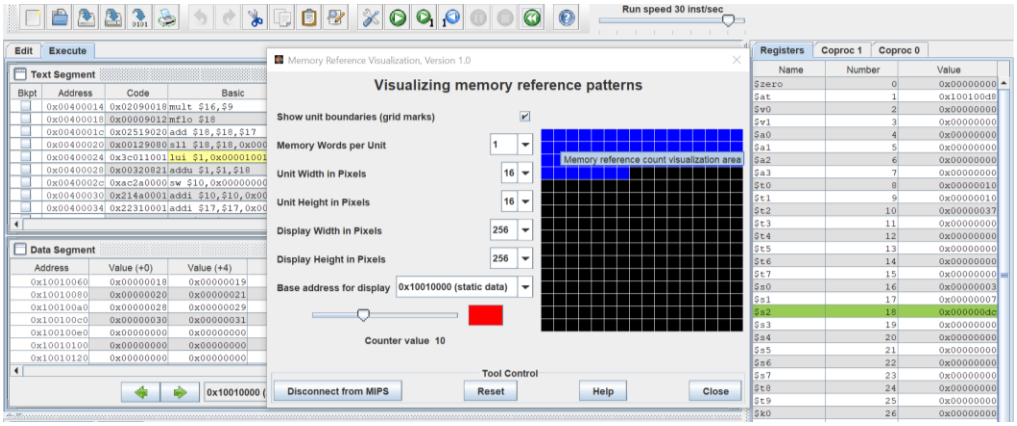
⇨ Các block không bị thay thế trong suốt chương trình

⇨ Chỉ có lần truy cập đầu tiên với mỗi block bị miss

⇨ Cache hit rate: 15/16 (≈ 94%)

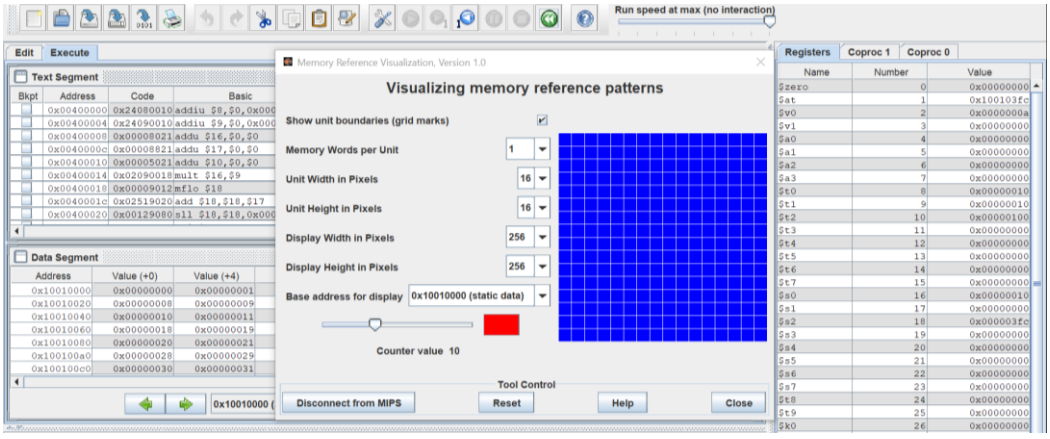
Part 2: The Memory Reference Visualization tool

1. Với file **`row-major.asm`**

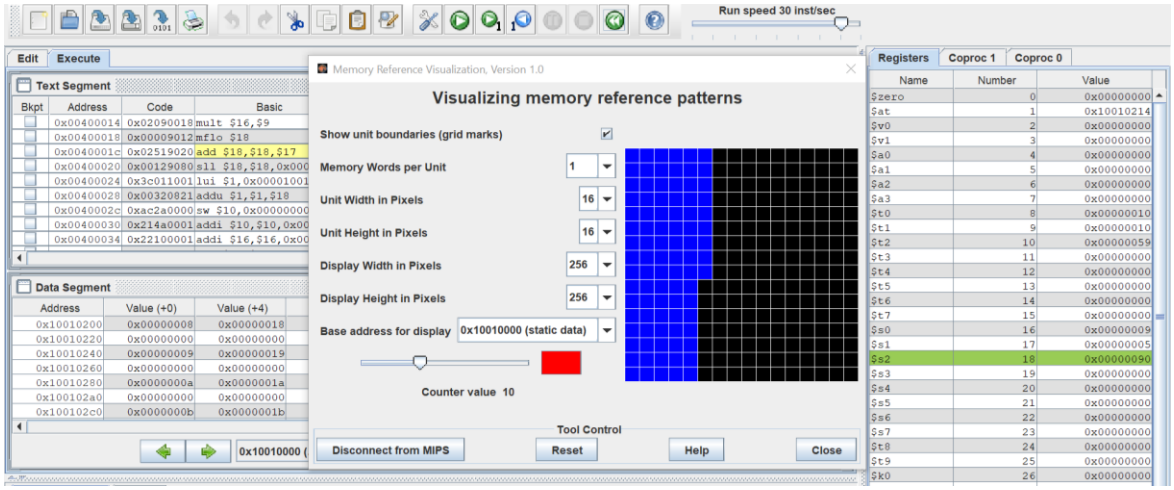


⇨ Bộ nhớ được truy cập bằng cách duyệt qua từng cột của lần lượt các hàng 1, 2, 3, …

⇨ Khi chạy xong toàn bộ chương trình toàn bộ các phần tử được truy cập và mỗi phần tử được truy cập đúng một lần

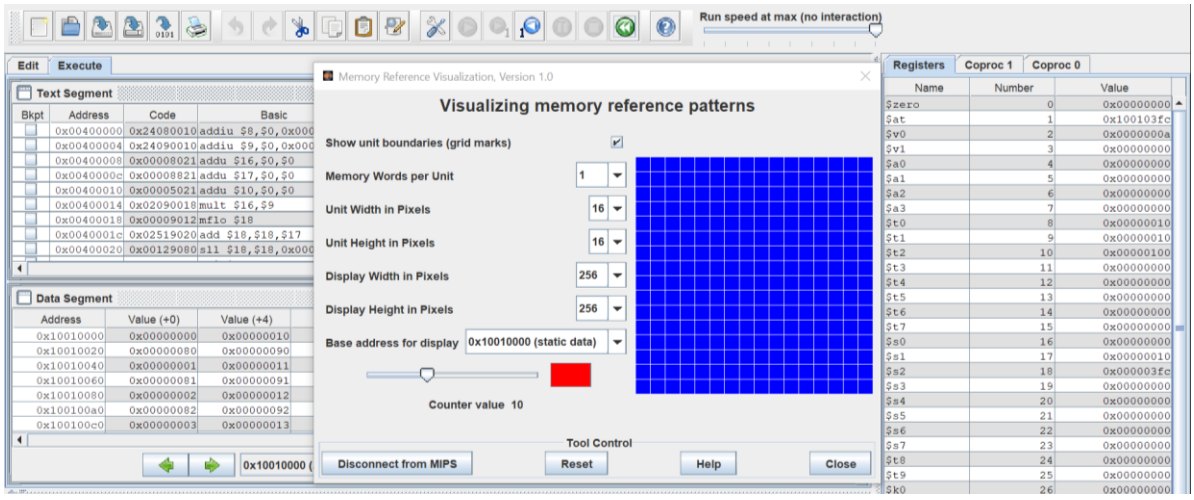


2. Với file **`column-major.asm`**



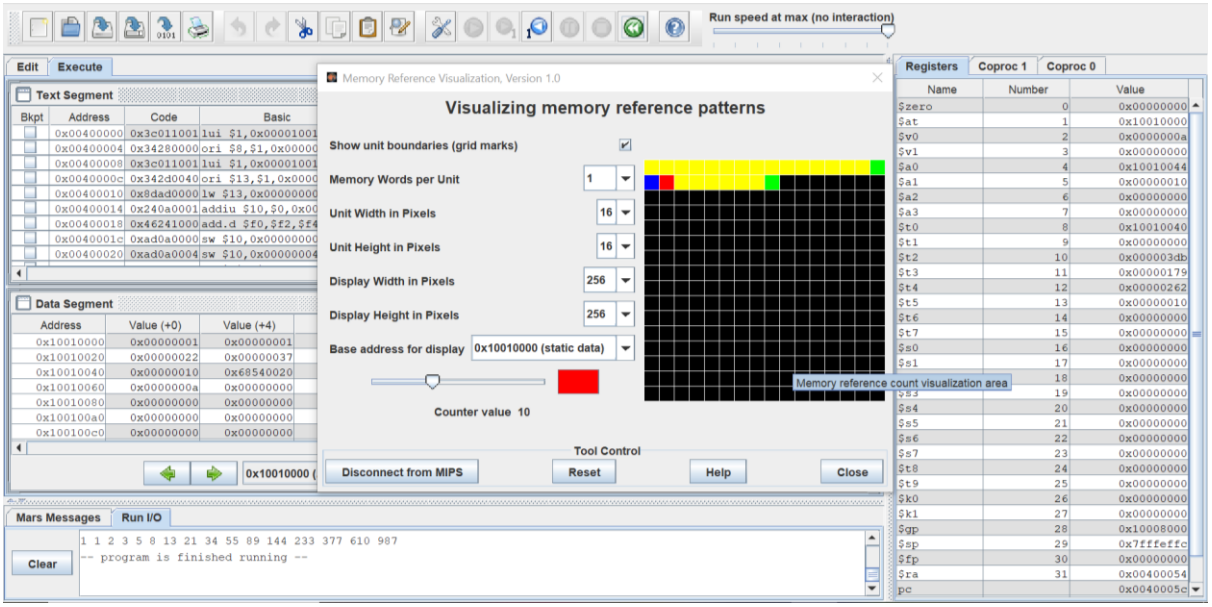
⇨ Bộ nhớ được truy cập bằng cách duyệt qua từng hàng của lần lượt các cột 1, 2, 3, …

⇨ Khi chạy xong toàn bộ chương trình toàn bộ các phần tử được truy cập và mỗi phần tử được truy cập đúng một lần



3. Với file **`fibonacci.asm`**

- Khi chạy toàn bộ chương trình:



- Khi chỉ chạy phần tính dãy fibonaci (bỏ qua print):