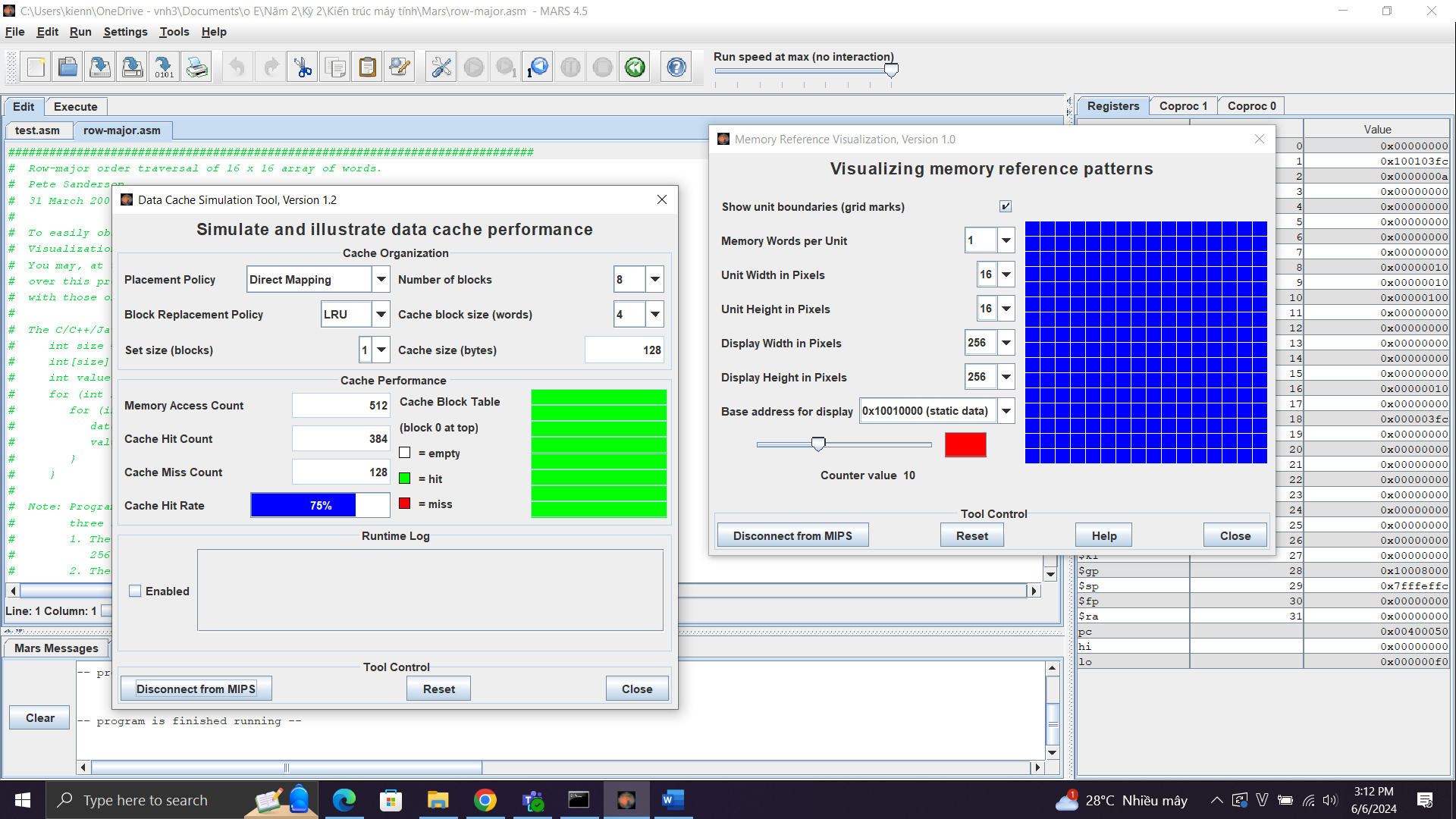
**BÁO CÁO THỰC HÀNH**

**KIẾN TRÚC MÁY TÍNH LAB**

**Nguyễn Trung Kiên – 20226110**

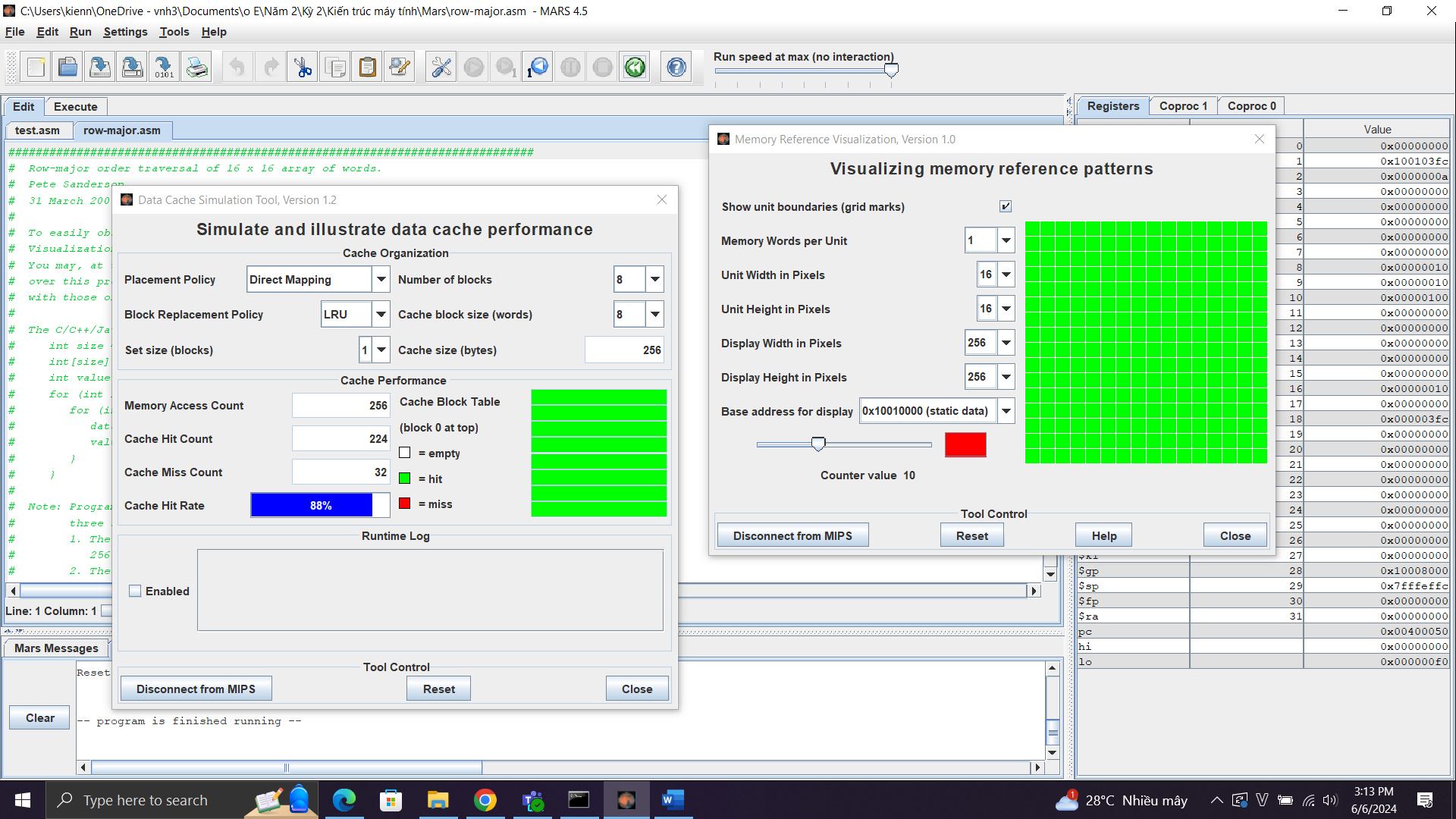
**Assignment 1:**

1. **Kết quả cuối cùng khi thực hiện từ bước 1 đến bước 7:**

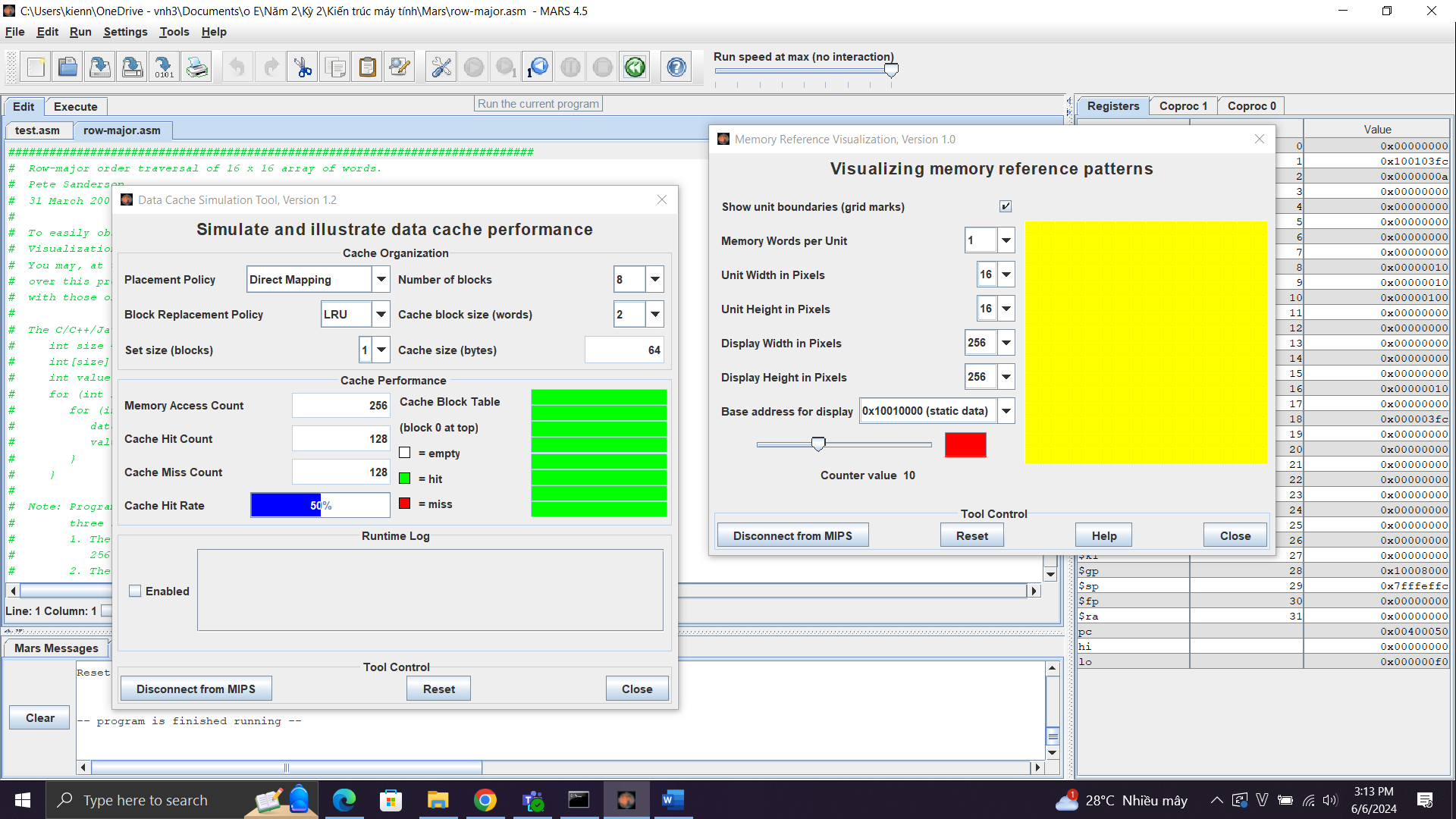


- Tỉ lệ cache hit rate cuối cùng: 75%. Vì với mỗi lần bỏ lỡ, một khối 4- word được ghi vào bộ đệm. Trong một đường truyền chính theo hàng, các phần tử của ma trận là được truy cập theo cùng thứ tự chúng được lưu trữ trong bộ nhớ. Do đó, mỗi lần bỏ lỡ bộ đệm là theo sau là 3 lần truy cập vì 3 phần tử tiếp theo được tìm thấy trong cùng một khối bộ đệm. Tiếp theo là một lỗi khác khi ma trận trực tiếp ánh xạ tới khối bộ đệm tiếp theo và sau đó lặp đi lặp lại chính nó. Vì vậy, 3 trong số 4 lần truy cập bộ nhớ sẽ được giải quyết trong bộ đệm.

- Khi tăng blocksize từ 4 lên 8 thì tỉ lệ Cache hit rate là 88%

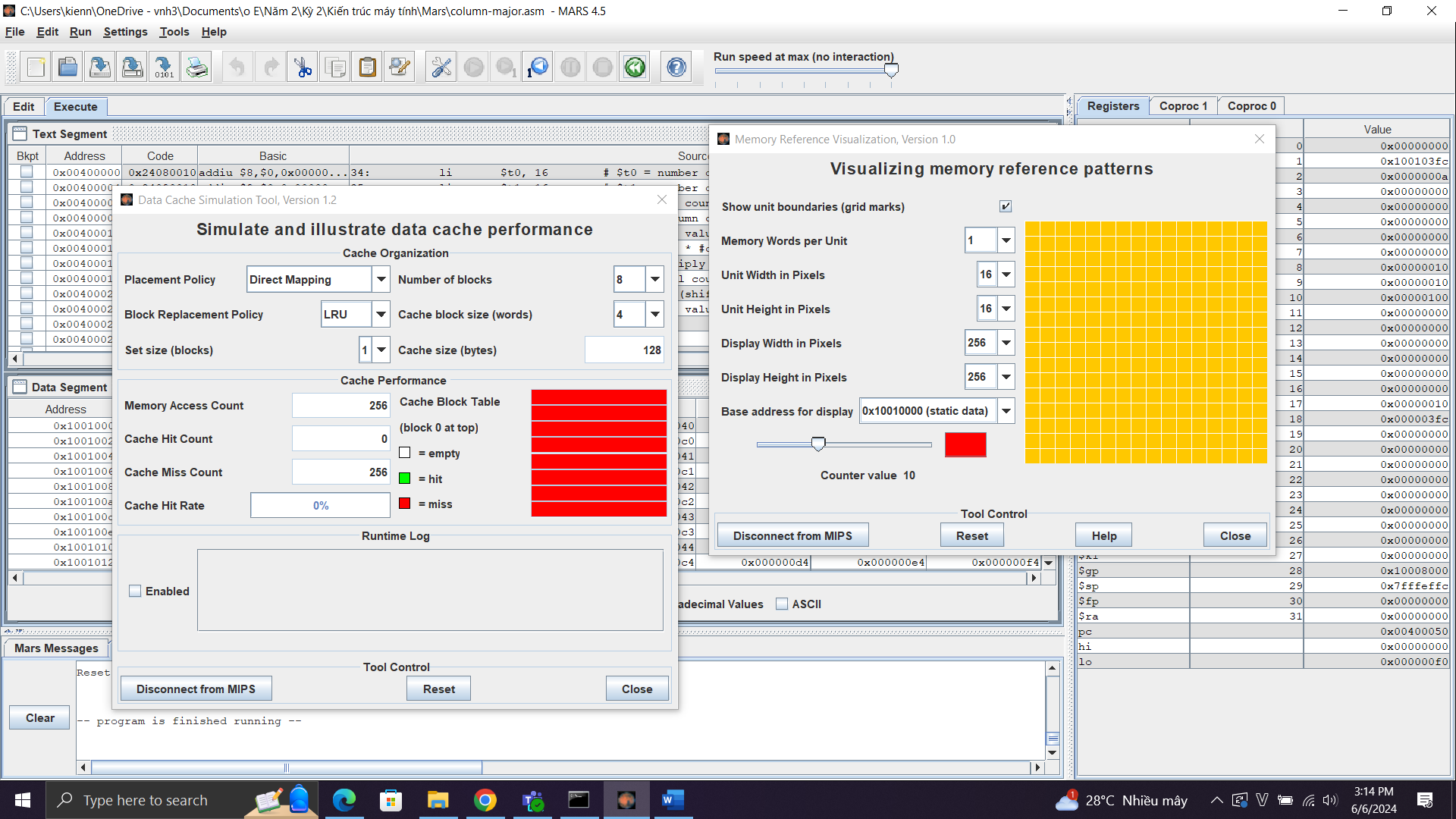


- Khi giảm blocksize từ 4 xuống 2 thì thỉ lệ Cache hit rate còn 50%



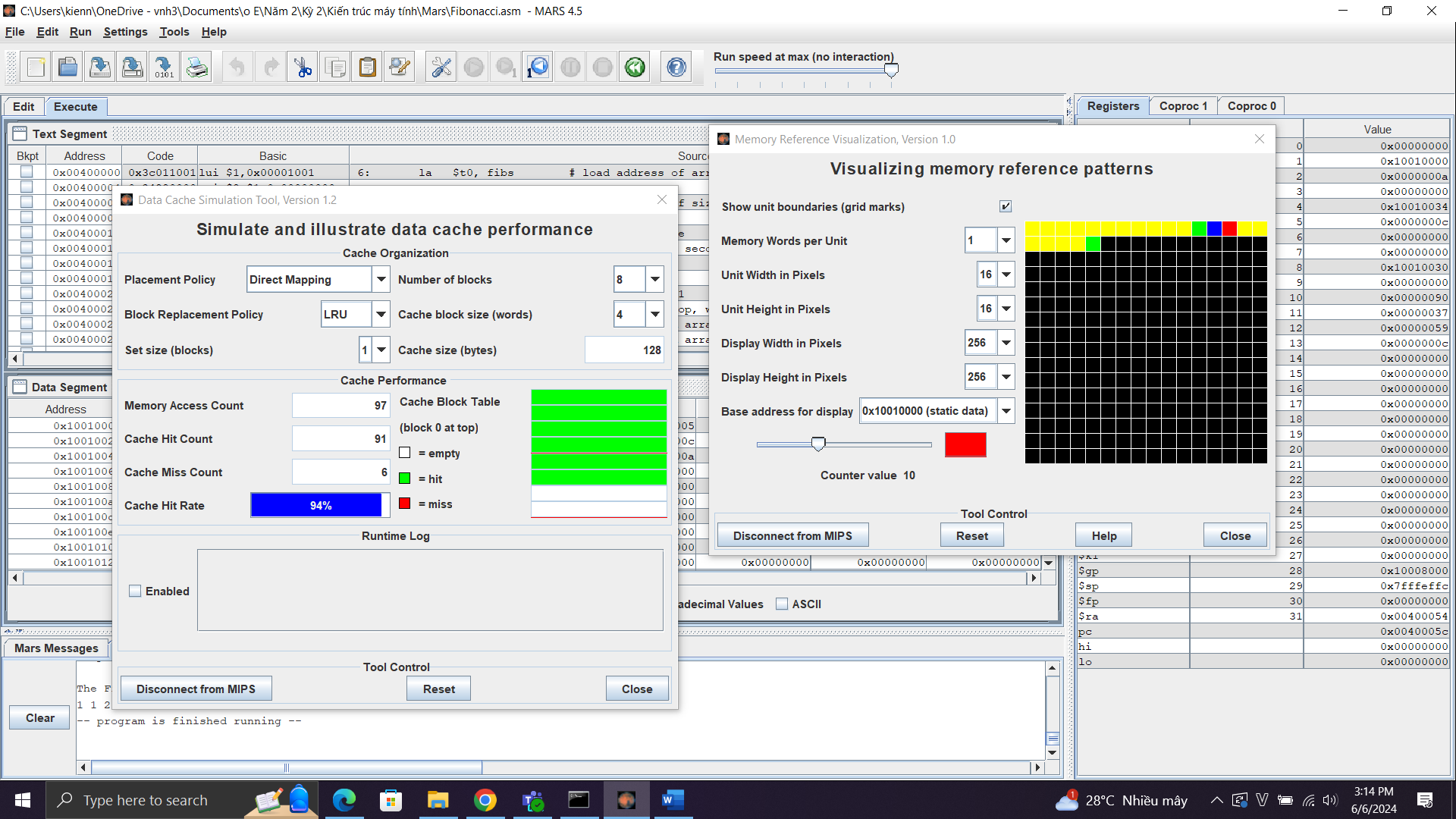
- Hiệu suất bộ đệm cũng bằng tỉ lệ Cache hit rate và bằng 75%

- Trường hợp chạy chương trình column:



- Khi chạy chương trình column, tỉ lệ Cache hit rate là 0%, suy ra hiệu suất bộ nhớ đệm cũng là 0%

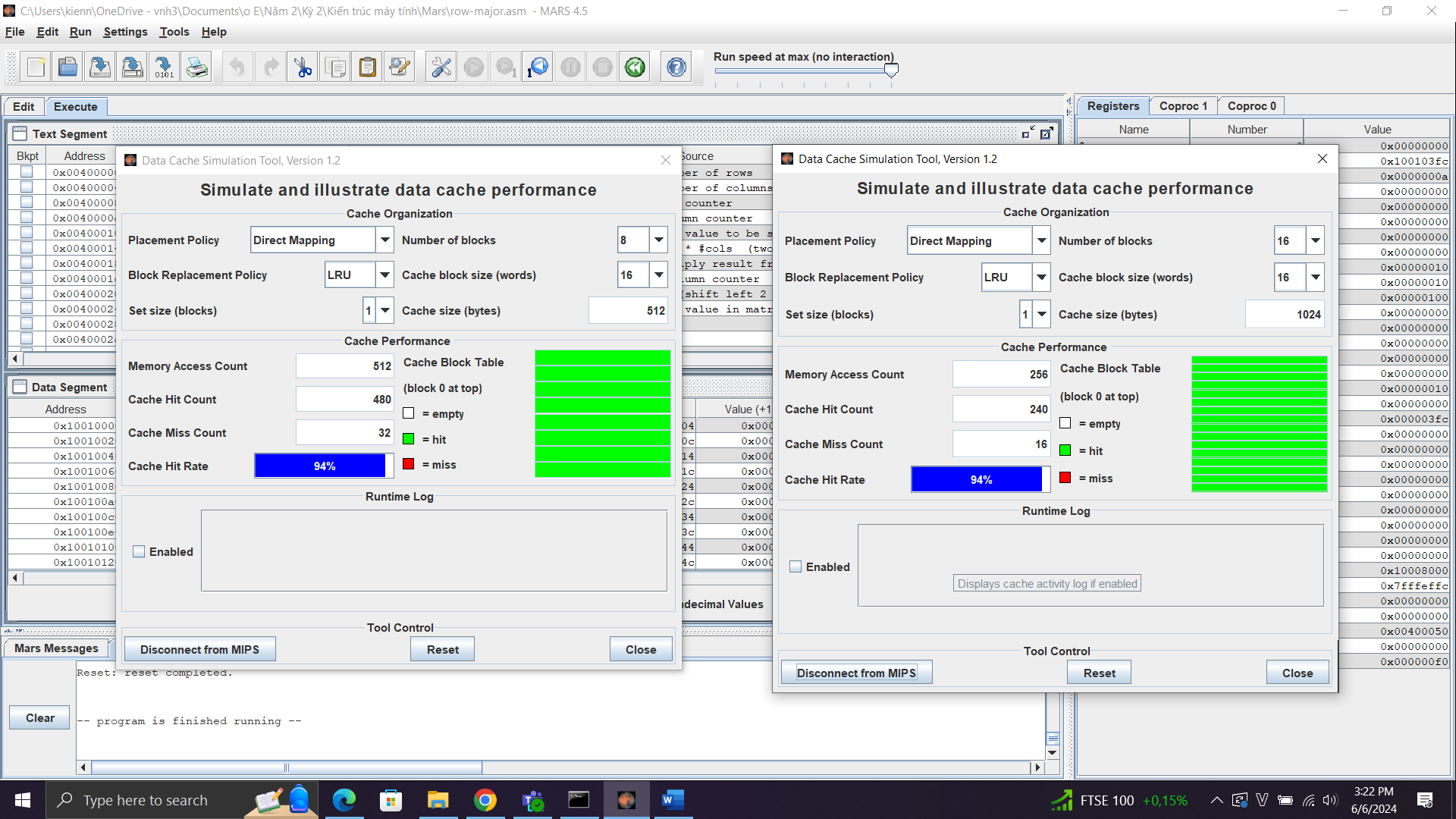
- Trường hợp chạy chương trình Fibonacci:



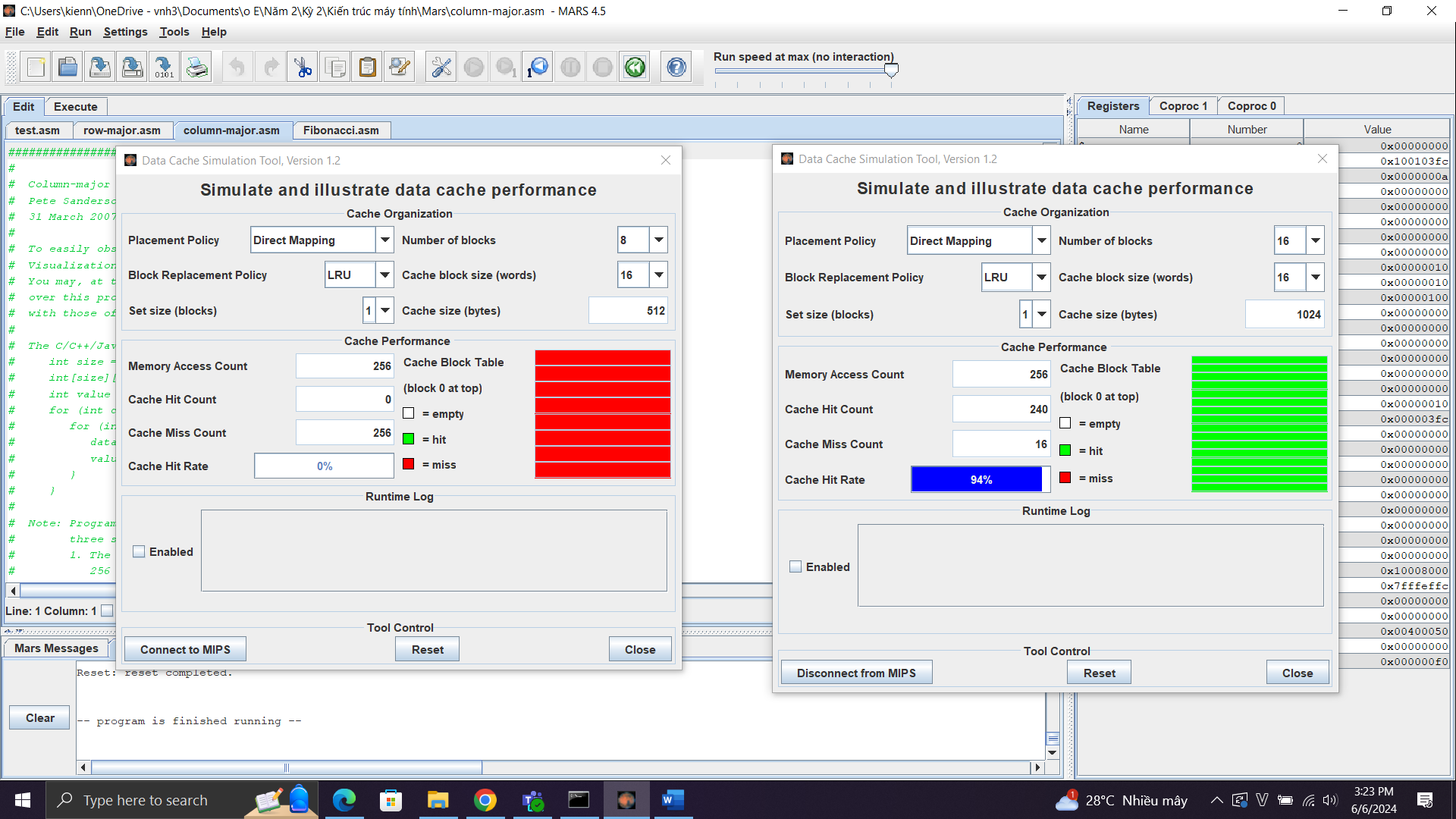
- Khi chạy chương trình Fibonacci, hiệu suất bộ nhớ đệm là 94%

1. **Thực hiện từ bước 13 đến bước 15 (cửa sổ data cache bên trái là bản gốc, bên phải là bản mới):**

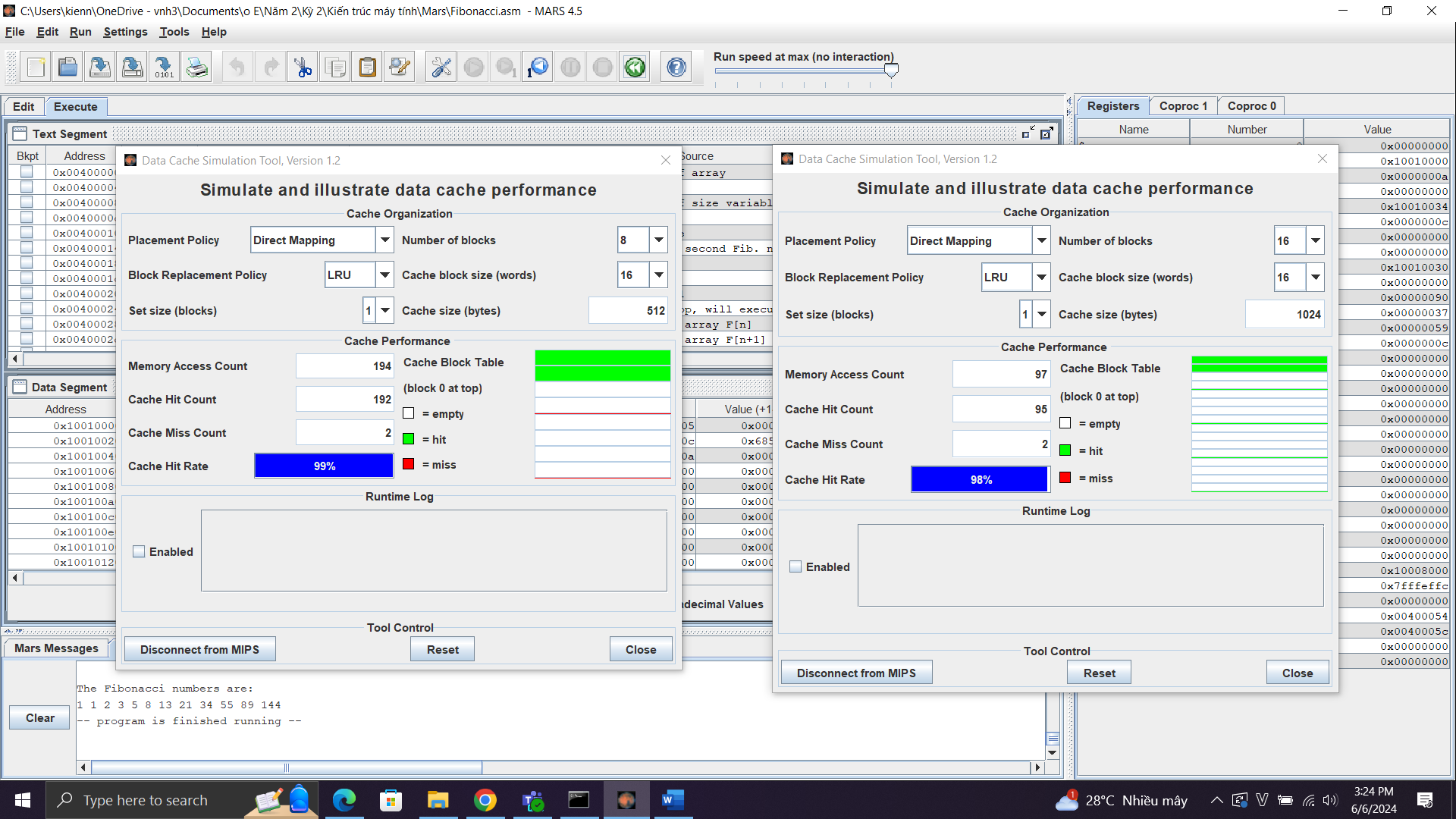
- Khi chạy chương trình row:



- Khi chạy chương trình column:



- Khi chạy chương trình Fibonacci:



* Trả lời bước 15:

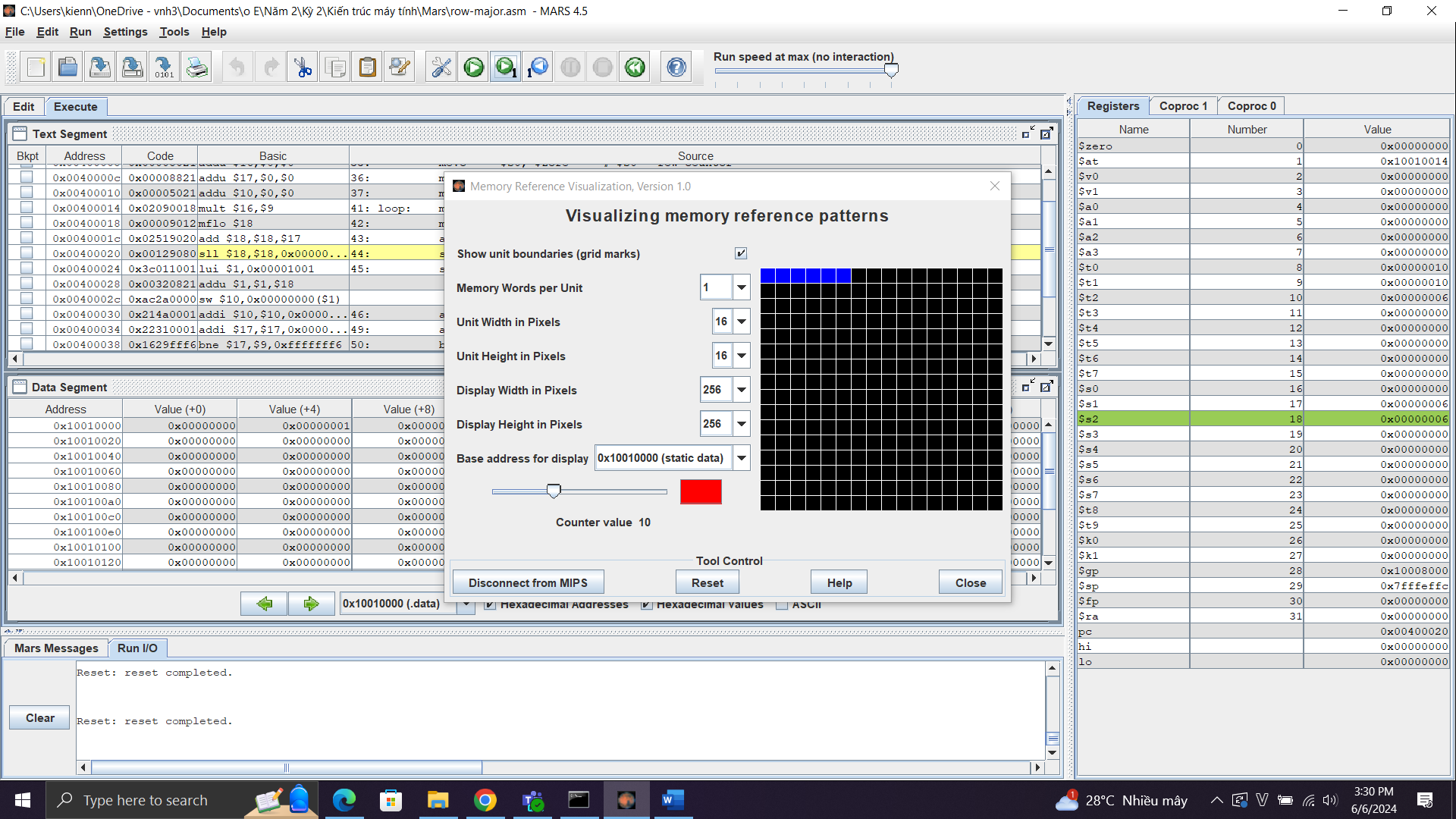
+ Hiệu suất bộ nhớ đệm của phiên bản công cụ gốc: Hiệu suất của phiên bản công cụ gốc **không hiệu quả**. Kích thước khối 16 không mang lại lợi ích bởi vì vẫn chỉ có một lần truy cập vào mỗi khối trước khi khối đó được thay thế bằng khối mới.

+ Hiệu suất bộ nhớ đệm của phiên bản công cụ thứ hai: Hiệu suất của phiên bản công cụ thứ hai **rất cao**. Ở điểm này, toàn bộ ma trận sẽ vừa vặn trong bộ nhớ đệm, do đó một khi một khối được đọc vào nó sẽ không bao giờ được thay thế. Chỉ có lần truy cập đầu tiên vào một khối mới là một lần trượt.

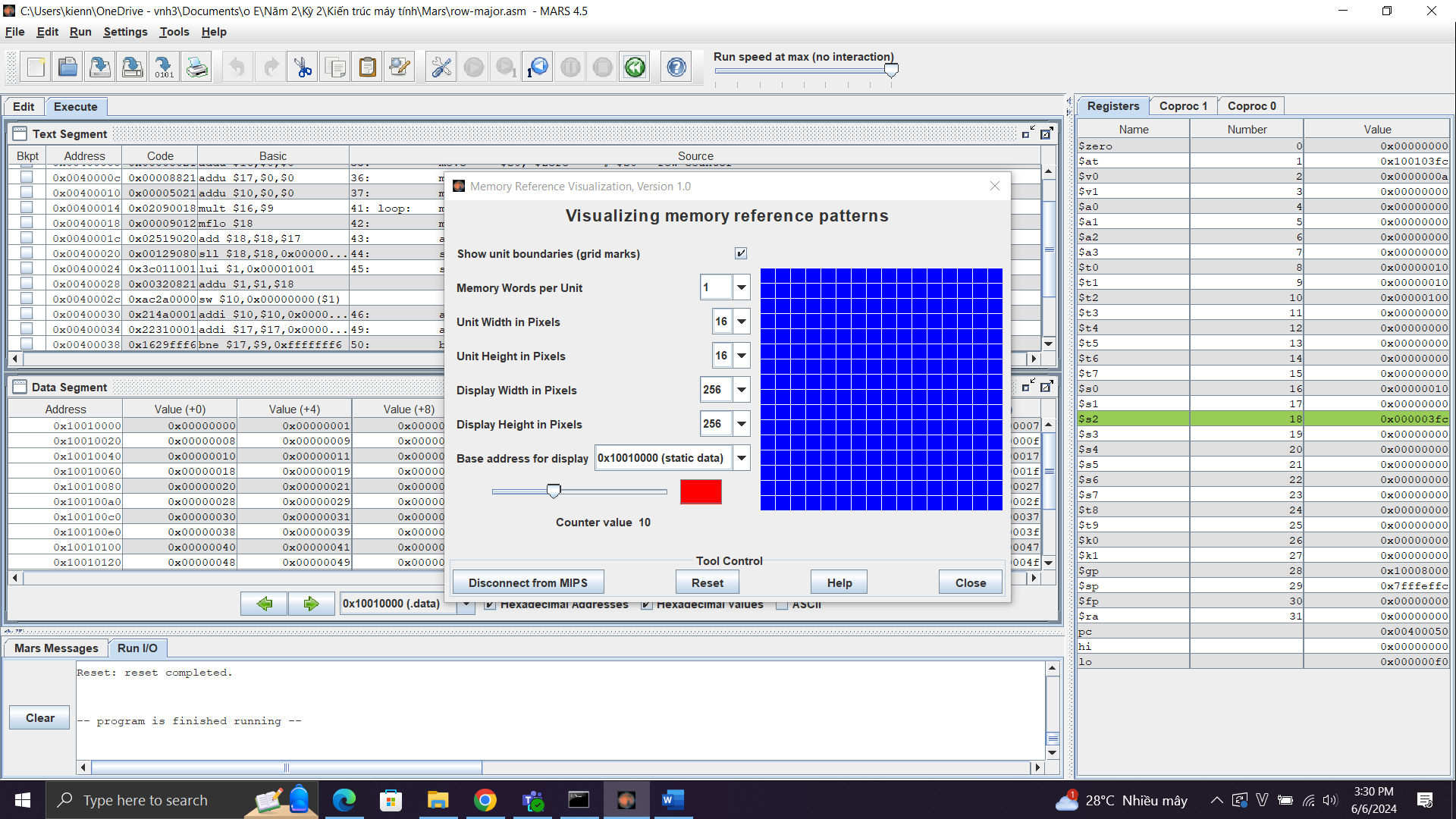
**Assignment 2:**

**1) Với chương trình row:**

- Khi đang thực thi:

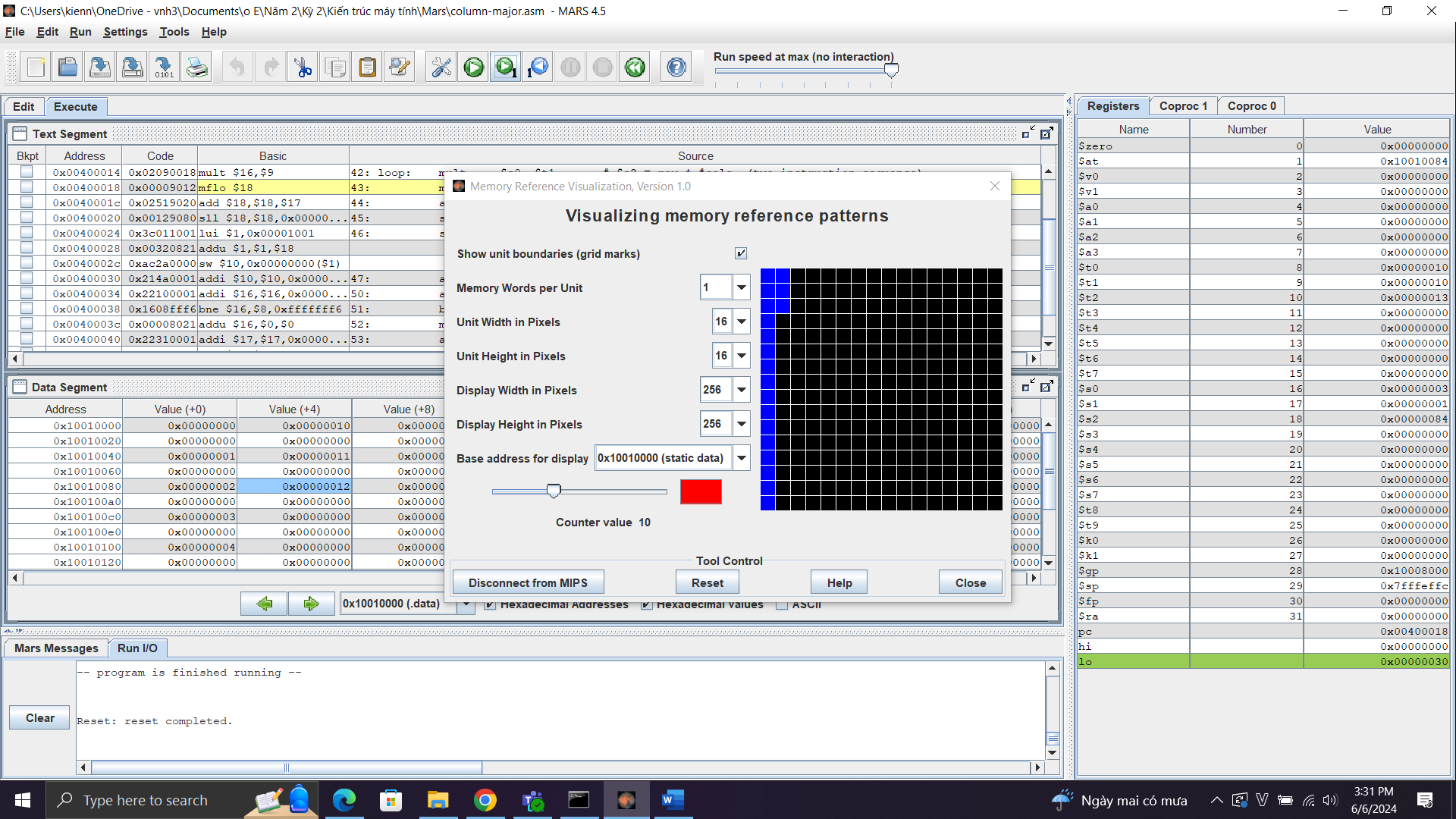


- Khi thực thi xong:

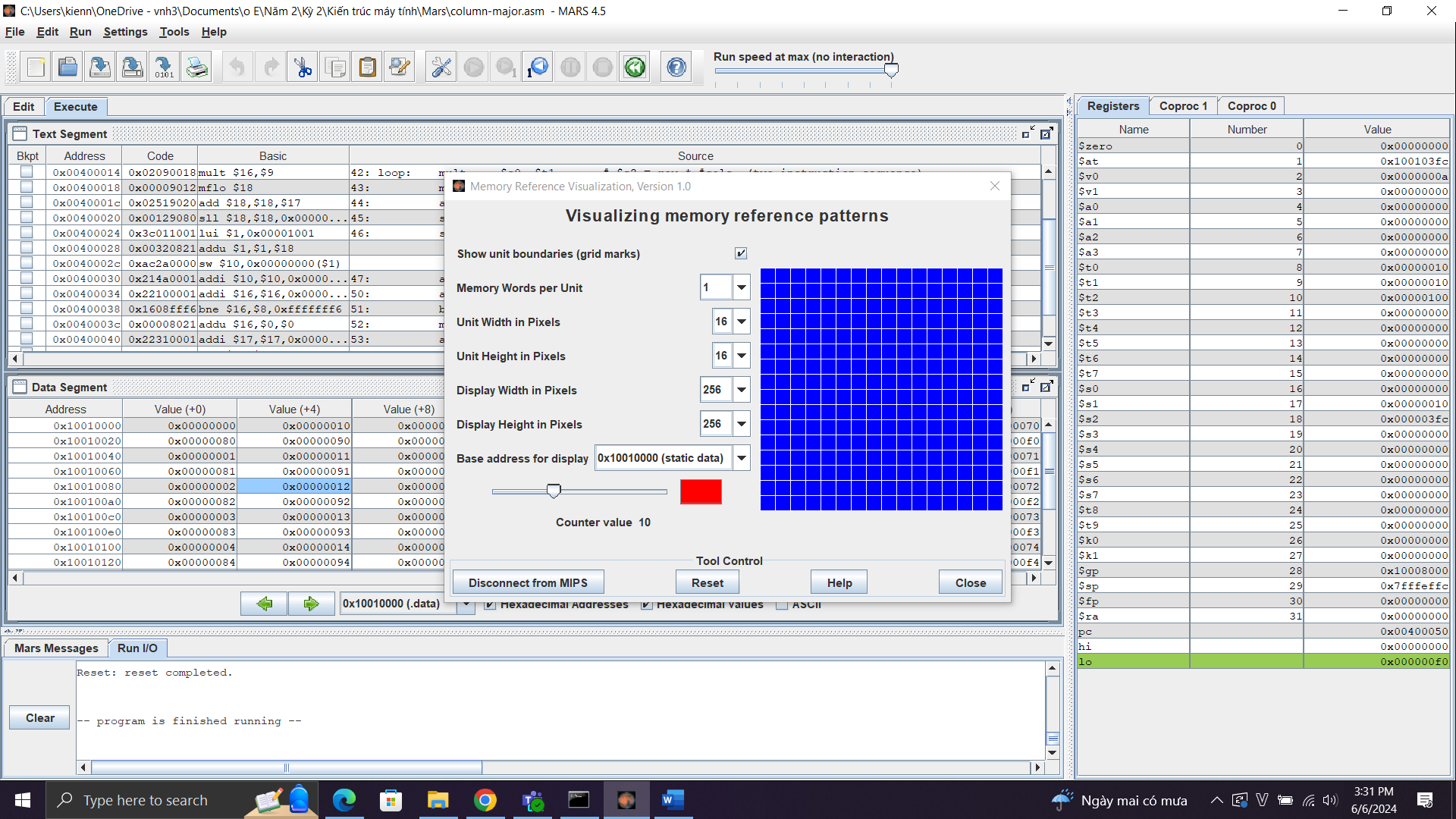


**2) Với chương trình column:**

- Khi đang thực thi:

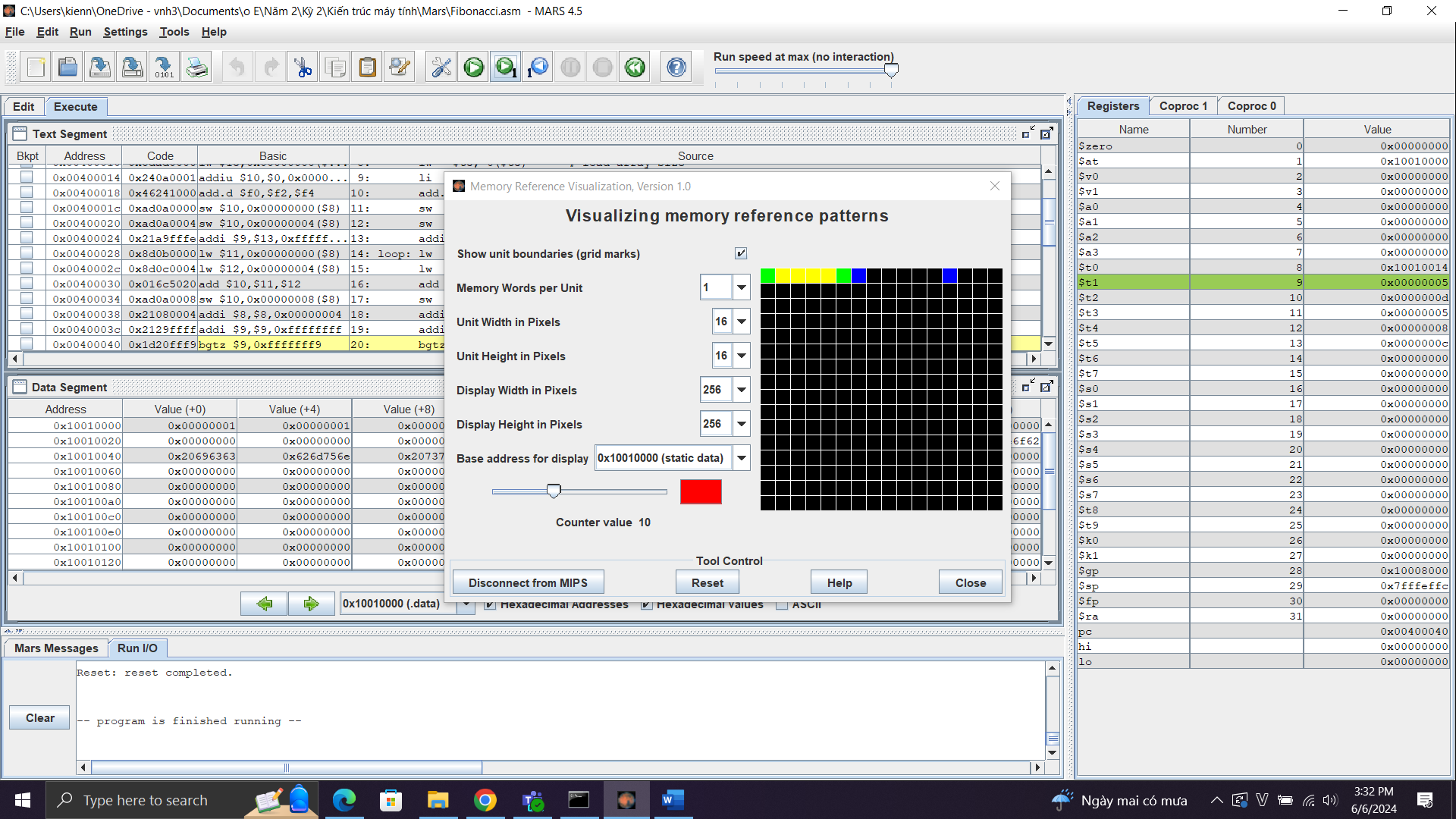


- Khi thực thi xong:



**3) Với chương trình Fibonacci:**

- Khi đang thực thi:



- Khi thực thi xong:

