Họ và tên: Trần Quang Nam Trường Đại Học Bách Khoa Hà Nội

MSSV: 20184161 Viện CNTT&TT

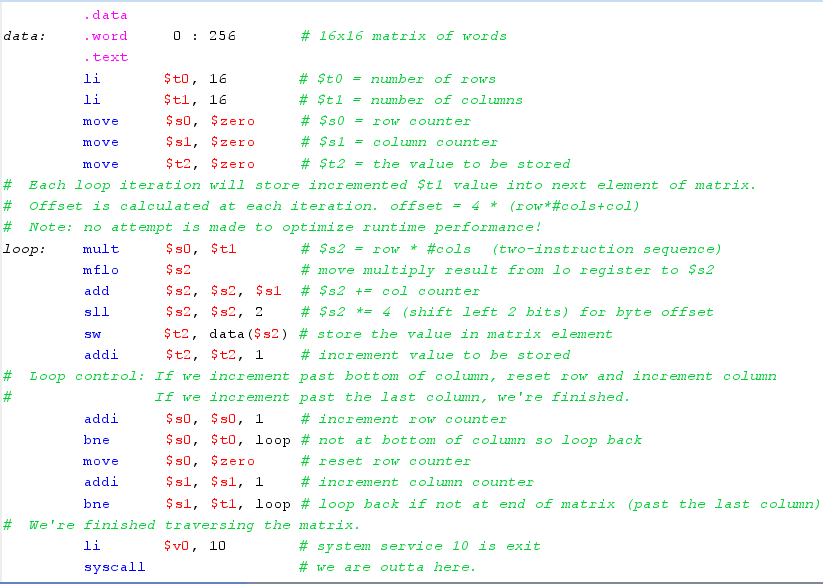
**BẢN BÁO CÁO THỰC HÀNH**

**-----TUẦN 01-----**

1. **NỘI DUNG THỰC HÀNH**

* Đọc hiểu file column-major.asm.
* Chạy trên phần mềm MARS.
* Viết báo cáo phân tích, giải thích các kết quả thu được.

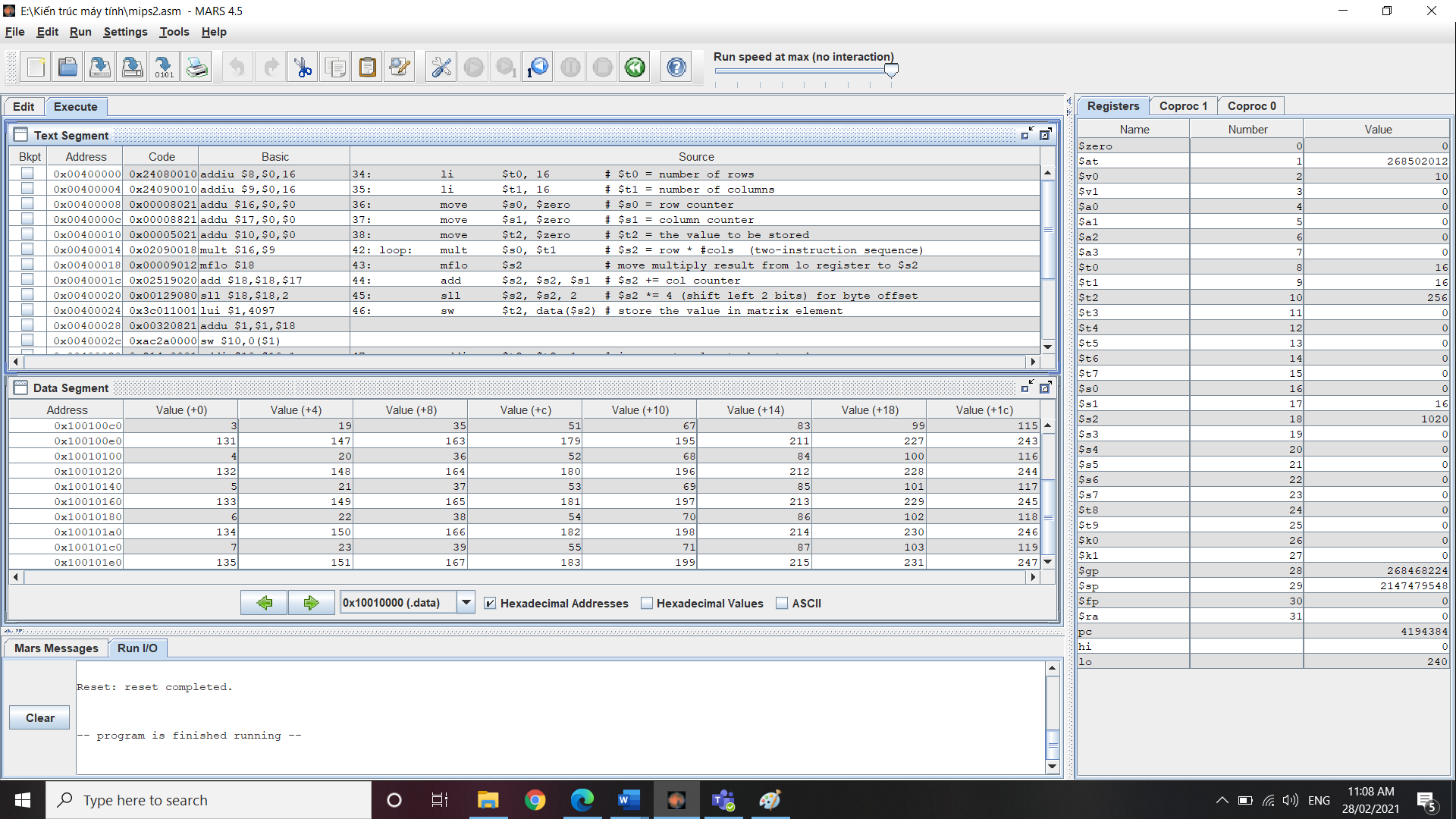
1. **BÁO CÁO**
   1. **Nội dung file column-major.asm**



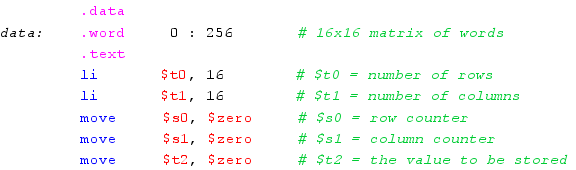
Đây là một đoạn chương trình hợp ngữ thực hiện chức năng khởi tạo và lưu trữ dữ liệu của ma trận vuông 16 x 16.

Cách duyệt các phần tử ma trận là cách duyệt mảng 1 chiều có 256 phần tử, mỗi phần tử là 1 dữ liệu “word”.

* 1. **Chạy trên phần mềm MARS**

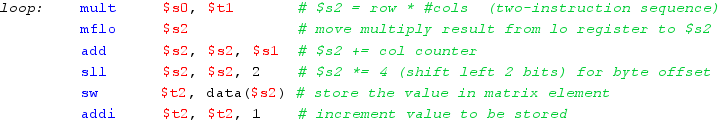


* 1. **Phân tích, giải thích**
* **Khai báo**



|  |  |  |
| --- | --- | --- |
| Câu lệnh | Ý nghĩa | Chú thích |
| data: .word 0 : 256 | Khai báo và khởi tạo 256 phần tử của ma trận, mỗi phần tử là một dữ liệu “word” | Dán nhãn data |
| li $to, 16 | Gán t0 = 16 | t0 là số hàng |
| li $t1, 16 | Gán t1 = 16 | t1 là số cột |
| move $s0, $zero | Gán s0 = 0 | s0 là biến đếm số hàng của ma trận, có giá trị ban đầu = 0 |
| move $s1, $zero | Gán s1 = 0 | s1 là biến đếm số cột của ma trận, có giá trị ban đầu = 0 |
| move $t2, $zero | Gán t2 = 0 | t2 là biến tạm thời lưu trữ dữ liệu của một phần tử ma trận |

* **Vòng lặp:**



|  |  |  |  |
| --- | --- | --- | --- |
| Câu lệnh | | Ý nghĩa | Chú thích |
| loop: mult $s0, $t1 | | Thực hiện phép nhân: s0\*t1, giá trị thu được lưu vào thanh ghi hi/lo | Dán nhãn loop |
| mflo $s2 | | Chuyển giá trị từ thanh ghi lo vào thanh ghi $s2 | s2 là biến đếm của mảng 1 chiều có 256 phần tử, mảng 1 chiều này đại diện cho ma trận vuông 16x16, s2 được tính bằng công thức:  s2 = s0\*16 + s1 (\*) |
| add $s2, $s2, $s1 | | Thực hiện phép cộng: s2=s2+s1 |
| sll $s2, $s2, 2 | | Thực hiện phép dịch 2 bit sang trái, s2 = s2 \* 4 (\*\*) | Gán giá trị của t2 vào phần tử thứ s2 của mảng 1 chiều |
| sw $t2, data($s2) | lui $at, 0x00001001 | Gán địa chỉ của phần tử đầu tiên của mảng vào thanh ghi tạm thời $at |
| addu $at, $at, $s2 | Thực hiện phép cộng:  at = at + s2 |
| sw $t2, 0($at) | Gán giá trị của t2 vào ô nhớ có địa chỉ được ghi trong thanh ghi $at (offset = 0) |
| addi $t2, $t2, 1 | | Thực hiện phép cộng:  t2 = t2 + 1 | Tạo giá trị mới cho phần tử tiếp theo |

Giải thích:

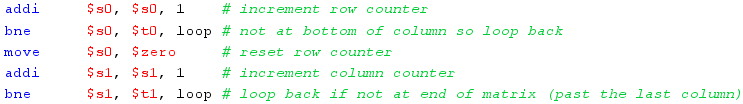
(\*) Cho ma trận vuông 16x16 có 256 phần tử, khi lưu trữ dùng mảng 2 chiều có dạng a[16][16]. Vì vậy, khi đổi sang mảng 1 chiều a[256] ta có công thức chuyển đổi phần tử tương ứng được sử dụng trong đoạn chương trình như sau:

a[i][j] = a[i\*16 + j]

(\*\*) Mỗi phần tử của mảng 1 chiều có kiểu dữ liệu “word” được lưu trong 4 byte bộ nhớ. Vì vậy, địa chỉ của phần tử liền sau bằng địa chỉ của phần tử trước đó cộng 4

Công thức tổng quát: &a[n] = &a[0] + 4\*n

* **Điều kiện vòng lặp:**



|  |  |  |
| --- | --- | --- |
| Câu lệnh | Ý nghĩa | Chú thích |
| addi $s0, $s0, 1 | Thực hiện phép cộng:  s0 = s0 + 1 | Tăng biến đếm hàng s0 |
| bne $s0, $t0, loop | So sánh s0 với t0. Nếu s0 khác t0 thì thực hiện vòng lặp loop |  |
| move $s0, $zero | Đặt lại giá trị của s0 về 0 | Sau khi thực hiện xong 1 vòng lặp loop thì một cột của ma trận đã được nhập hết, vì vậy cần đặt lại biến đếm hàng về 0 |
| addi $s1, $s1, 1 | Thực hiện phép cộng:  s1 = s1 + 1 | Tăng biến đếm cột s1 |
| bne $s1, $t1, loop | So sánh s1 với t1. Nếu s1 khác t1 thì thực hiện vòng lặp loop | Sau khi nhập hết giá trị các phần tử trong 1 cột thì chuyển sang cột tiếp theo và tiếp tục lại vòng lặp |

* Kết thúc chương trình

