

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



## KIÊN TRÚC MÁY TÍNH (CO2007)

---

### BÁO CÁO BÀI TẬP LỚN NHÓM 17 - HK222

#### Đề 8: In phổ sao

---

GVHD: Nguyễn Xuân Minh  
SV thực hiện: Phan Đức Đạt – 2113152  
Phạm Tiến Đạt – 2113155



## Mục lục

<b>1</b>	<b>Giải pháp hiện thực</b>	<b>2</b>
<b>2</b>	<b>Các lệnh được sử dụng trong chương trình</b>	<b>2</b>
2.1	Nhóm lệnh số học . . . . .	2
2.2	Nhóm lệnh nhánh/nhảy . . . . .	2
2.3	Nhóm lệnh memory-instruction . . . . .	3
<b>3</b>	<b>Giải thuật bằng sơ đồ</b>	<b>3</b>
<b>4</b>	<b>Kết quả chạy thử</b>	<b>3</b>
4.1	Test Case 1 . . . . .	3
4.2	Test Case 2 . . . . .	4
4.3	Test Case 3 . . . . .	5
4.4	Test Case 4 . . . . .	6
4.5	Test Case 5 . . . . .	6
4.6	Test Case 6 . . . . .	7

## 1 Giải pháp hiện thực

- Đề bài: Viết chương trình in phổ sao như hình bên dưới:



Hình 1: Kết quả hiện thực của đề mẫu

Chuỗi 40 ký số ASCII đọc từ tập tin STRING.TXT trên đĩa.

- Giải pháp:

+ Bước 1: Đọc 40 ký số ASCII từ file STRING.txt trên đĩa vào một string s có space là 40.

(string s đó được tạo trong MARS MIPS)

+ Bước 2: Tạo 9 string star1 đến star 9 tương ứng với 9 hàng trong phổ sao. Hàng đầu tiên là hàng 9, hàng cuối cùng là hàng 1

+ Bước 3: Ở mỗi string star, tiến hành chạy vòng for i để thực hiện 2 hành động:

. Hành động 1: Nếu ký tự thứ i trong string s bằng với số hàng đang được thực thi thì ký tự thứ i trong string star của hàng này sẽ được gán bằng ký tự '\*', còn không thì nó được gán bằng ký tự ' ' (dấu cách).

. Hành động 2: Gán ký tự thứ i của string s trên bằng số hàng - 1 (Hành động 2 chỉ xảy ra khi ký tự thứ i trong string s bằng với số hàng đang được thực thi)

+ Bước 4: Sau khi thực hiện hết 9 hàng của phổ sao, tiến hành in phổ sao ra màn hình và kết thúc chương trình.

## 2 Các lệnh được sử dụng trong chương trình

### 2.1 Nhóm lệnh số học

Lệnh	Cú pháp	Hoạt động
addi	addi \$t0, \$t1, 3	Cộng giá trị thanh ghi \$t1 với số 3, tổng đưa vào thanh ghi \$t0. Thông báo kết quả bị overflow (nếu có).
addiu	addiu \$t0, \$t1, 1	Cộng giá trị thanh ghi \$t1 với số 1, tổng đưa vào thanh ghi. Nhưng không xét kết quả có bị overflow hay không.

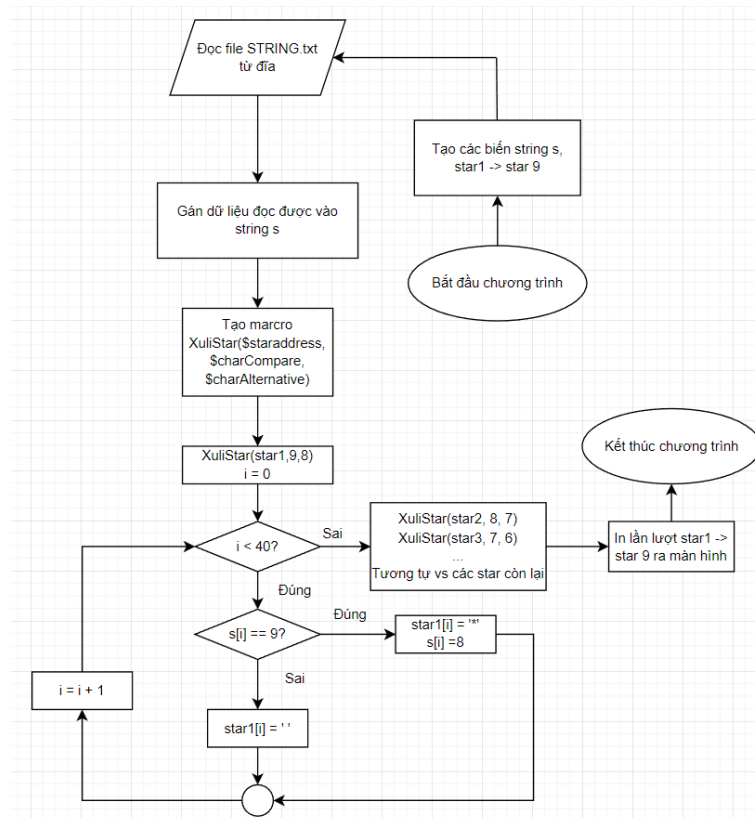
### 2.2 Nhóm lệnh nhánh/nhảy

Lệnh	Cú pháp	Hoạt động
beq	beq \$t0, \$t1, label	Nếu \$t0 bằng \$t1 thì nhảy đến label.
j	j label	Nhảy vô điều kiện đến label.

## 2.3 Nhóm lệnh memory-instruction

Lệnh	Cú pháp	Hoạt động
lb	lb \$t0, \$t1, label	Nếu \$t0 bằng \$t1 thì nhảy đến label.
li	li \$t0, 14	Đưa một số tức thời 14 (32 bits) vào thanh ghi \$t0.
la	la \$a0, label	Lưu địa chỉ của label vào \$a0.
sb	sb \$t1, 8(\$t0)	Lấy byte thấp nhất của trong thanh ghi \$t0 lưu vào byte thấp nhất của \$t1 có địa chỉ được tính bằng giá trị thanh ghi \$t0 cộng với offset 8 (offset được mở rộng có dấu thành số 32 bits trước khi cộng).
move	move \$t1, \$t0	Sao chép/di chuyển giá trị từ thanh ghi \$t0 sang thanh ghi \$t1.

## 3 Giải thuật bằng sơ đồ



Hình 3: Giải thuật bằng sơ đồ của đề 8: In phổ sao

## 4 Kết quả chạy thử

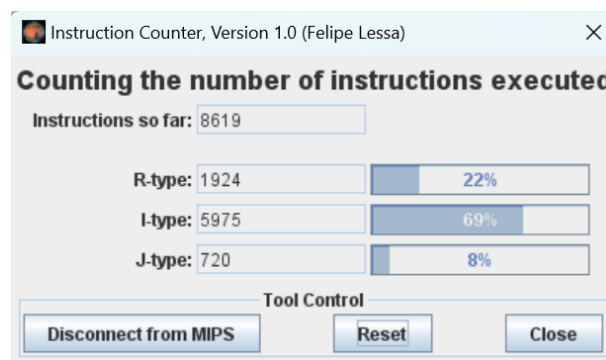
### 4.1 Test Case 1

- Input: 0010010023456700006543210000789870023417
- Kết quả chạy MARS MIPS:

0010010023456700006543210000789870023417											
					★						
					★★★						
★					★★★★★			★			
★★		★				★★★★★			★		
★★★			★★				★★★★★			★	
★★★★			★★★				★★★★★			★ ★	
★★★★★				★★★★				★★★★★			★ ★ ★
★★★★★★				★★★★★				★★★★★			★ ★ ★ ★
★	★	★★★★★★			★★★★★★				★★★★★		★★★★★

Hình 4.1.1: Kết quả chạy MARS MIPS của Test Case 1

- Tổng số lệnh thực thi:



Hình 4.1.2: Tổng số lệnh thực thi của Test Case 1

- Thời gian thực thi:  $CPUTime = \frac{InstructionCount \times CPI}{ClockRate} = \frac{8619 \times 1}{10^9} = 8619 \text{ (ns)}$

## 4.2 Test Case 2

- Input: 1028946758361846290232946324701947873643

- Kết quả chạy MARS MIPS:

```

1028946758361846290232946324701947873643

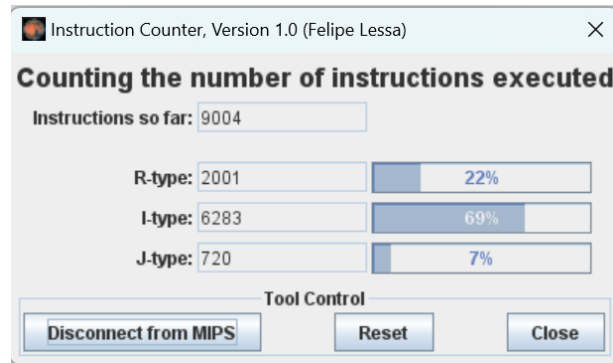
      *               *       *           *
    **             *   *     *         *   *
  ***          * *   *     *       *   *   ***
***   ***   * *   *   *   *       *   *   *   ***
**   ****   * *   *   *   *       *   *   *   ****
*****   *****   *   ****   *       ****   ****   *****
****   *****   ****   *   *   ****   ****   *****
***   *****   ****   ****   *****   *****
**   *****   ****   ****   *****   *****
*   *****   ****   ****   *****   *****

-- program is finished running --

```

Hình 4.2.1: Kết quả chạy MARS MIPS của Test Case 2

- Tổng số lệnh thực thi:

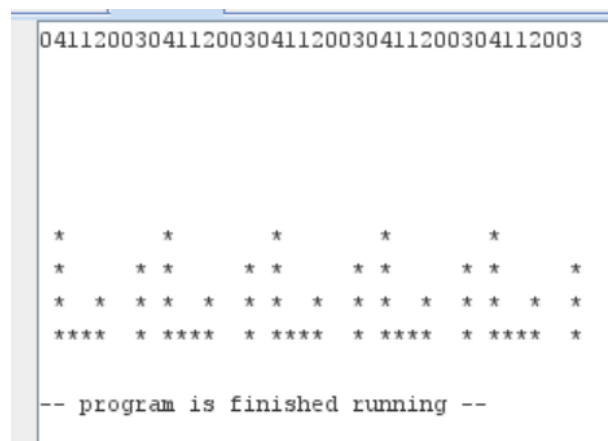


Hình 4.2.2: Tổng số lệnh thực thi của Test Case 2

- Thời gian thực thi:  $CPUTime = \frac{InstructionCount \times CPI}{ClockRate} = \frac{9004 \times 1}{10^9} = 9004 \text{ (ns)}$

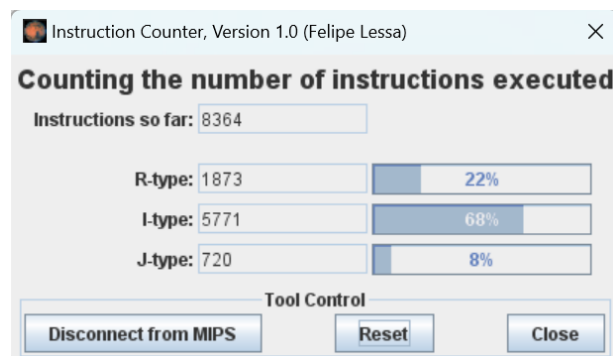
### 4.3 Test Case 3

- Input: 0411200304112003041120030411200304112003
- Kết quả chạy MARS MIPS:



Hình 4.3.1: Kết quả chạy MARS MIPS của Test Case 3

- Tổng số lệnh thực thi:



Hình 4.3.2: Tổng số lệnh thực thi của Test Case 3

- Thời gian thực thi:  $CPUTime = \frac{InstructionCount \times CPI}{ClockRate} = \frac{8364 \times 1}{10^9} = 8364 \text{ (ns)}$

#### 4.4 Test Case 4

- Input: 8537578733872344070509834507435075073756
- Kết quả chạy MARS MIPS:

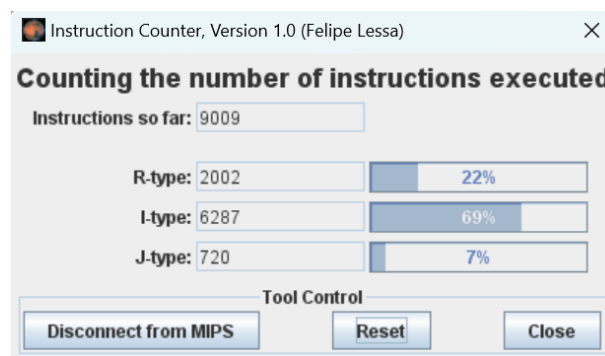
```

8537578733872344070509834507435075073756
          *
      *
*           *           *           *           *
*   *   **   *   *           *   *   *           *   *   *
*   *   **   *   *           *   *   *           *   *   *
***   ****   *   *           *   *   *   *   *   *   *   *
**   ****   *   *   *   *   *   *   *   *   *   *
*****   ****   *   *   *   *   *   *   *   *   *
*****   ****   *   *   *   *   *   *   *   *   *
*****   ****   *   *   *   *   *   *   *   *   *
-- program is finished running --

```

Hình 4.4.1: Kết quả chạy MARS MIPS của Test Case 4

- Tổng số lệnh thực thi:



Hình 4.4.2: Tổng số lệnh thực thi của Test Case 4

- Thời gian thực thi:  $CPUTime = \frac{InstructionCount \times CPI}{ClockRate} = \frac{9009 \times 1}{10^9} = 9009 \text{ (ns)}$

#### 4.5 Test Case 5

- Input: 2113155211315221131522113155211352211315
- Kết quả chạy MARS MIPS:

```

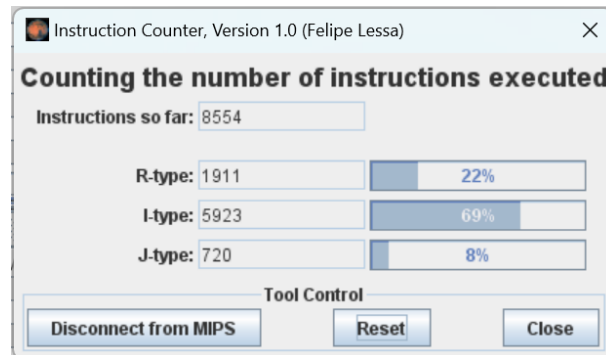
2113155211315221131522113155211352211315

          *      *      *      **      *      *
        **      *      *      **      *      *
       ***     **   **    ****   **   **   ***
    *   **   **   **   **   **   **   **   **   **
  *   **   ***   **   ***   **   ***   ****   **
*****
-- program is finished running --

```

Hình 4.5.1: Kết quả chạy MARS MIPS của Test Case 5

- Tổng số lệnh thực thi:

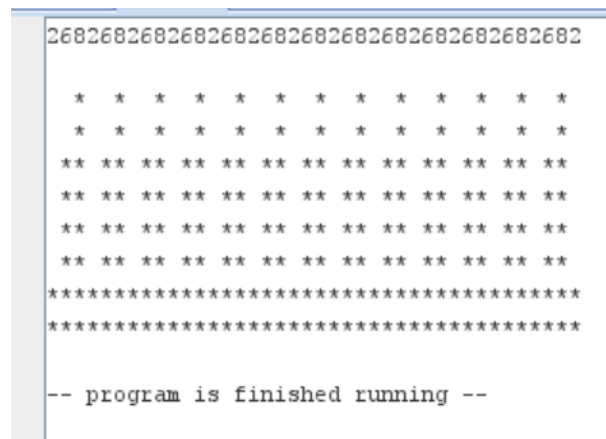


Hình 4.5.2: Tổng số lệnh thực thi của Test Case 5

- Thời gian thực thi:  $CPUTime = \frac{InstructionCount \times CPI}{ClockRate} = \frac{8554 \times 1}{10^9} = 8554 \text{ (ns)}$

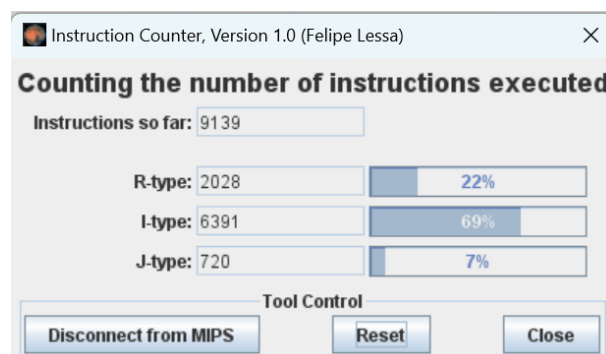
## 4.6 Test Case 6

- Input: 2682682682682682682682682682682682
- Kết quả chạy MARS MIPS:



Hình 4.6.1: Kết quả chạy MARS MIPS của Test Case 6

- Tổng số lệnh thực thi:



Hình 4.6.2: Tổng số lệnh thực thi của Test Case 6

- Thời gian thực thi:  $CPUTime = \frac{InstructionCount \times CPI}{ClockRate} = \frac{9139 \times 1}{10^9} = 9139 \text{ (ns)}$