

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



BÀI TẬP LỚN

KIẾN TRÚC MÁY TÍNH

GVHD: Trần Thanh Bình
SV thực hiện: Nguyễn Thành Đạt _1910113
Trương Vĩnh Phước _1910473
Lại Đức Anh Khoa _1910265

Tp. Hồ Chí Minh, Tháng 11/2020



Mục lục

1	Động cơ nghiên cứu	2
2	Mục tiêu	2
3	Yêu cầu	2
4	Nội dung báo cáo	2
4.1	Sử dụng hợp ngữ MIPS, viết thủ tục sắp xếp chuỗi theo thứ tự tăng dần bằng giải thuật Quick sort	2
4.2	Thống kê số lệnh, loại lệnh đã sử dụng	7
4.3	Tính và trình bày cách tính thời gian chạy của chương trình	10
5	Tài liệu tham khảo	12

1 Động cơ nghiên cứu

Kiến trúc tập lệnh (Instruction Set Architecture, viết tắt là ISA) là giao diện chính giữa phần cứng và phần mềm, là cái nhìn trừu tượng của phần cứng trên quan điểm phần mềm. Hiểu được kiến trúc tập lệnh giúp cho lập trình viên có thể hiểu được máy tính hoạt động như thế nào. Từ đó, ngoài khả năng truy cập vào phần cứng hệ thống, lập trình viên còn có thể đưa ra những phương án lập trình mang lại hiệu quả cao về thời gian xử lý và không gian lưu trữ.

2 Mục tiêu

Thông qua bài tập lớn lần này, sinh viên có thể hiểu rõ được kiến trúc tập lệnh MIPS (Microprocessor without Interlocked Pipelined Stages). Đồng thời sinh viên còn có thể sử dụng hợp ngữ MIPS để hiện thực những giải thuật đơn giản thường gặp, giúp cho việc sử dụng các ngôn ngữ lập trình cấp cao sẽ trở nên tối ưu và hiệu quả hơn.

3 Yêu cầu

- Sử dụng hợp ngữ MIPS, viết thủ tục sắp xếp chuỗi theo thứ tự tăng dần bằng giải thuật Quick sort
- Thống kê số lệnh, loại lệnh của chương trình của nhóm
- Tính và trình bày cách tính thời gian chạy của chương trình trên máy tính MIPS có tần số 2GHz

4 Nội dung báo cáo

4.1 Sử dụng hợp ngữ MIPS, viết thủ tục sắp xếp chuỗi theo thứ tự tăng dần bằng giải thuật Quick sort

a Lý thuyết:

Ý tưởng: Thuật toán Quick sort lựa chọn một phần tử của mảng cần sắp xếp làm pivot và chia mảng trên thành hai mảng con, một mảng nhỏ hơn và một mảng lớn hơn pivot. Tiếp tục thực hiện quá trình này trên các mảng con cho đến khi mảng được sắp xếp hoàn toàn.

Giải thuật:

- Hàm partition: chọn phần tử cuối cùng của mảng làm pivot, đưa tất cả các phần tử nhỏ hơn pivot về phía bên trái của mảng rồi đưa pivot vào đúng vị trí của nó, khi đó các phần tử lớn hơn pivot sẽ tự động được đưa về phía bên phải của mảng. Trả về vị trí đúng của pivot trong mảng.
- Hàm quickSort: gọi hàm partition cho mảng ban đầu, đánh dấu vị trí pivot. Tiếp tục thực hiện đối với mảng con bên trái và mảng con bên phải bằng cách gọi đệ quy hàm quickSort.

b Mã giả:

```
algorithm quickSort(array <int>, val low <int>, val high <int>)
```

```
Sorts the contiguous list using quick Sorts
```

```
Pre: low and high are valid positions in the list
```

```
Post: sorted list
```

```
    if low < high
        pivot_position = partition(array, low, high)
        quickSort(array, low, pivot_position - 1)
        quickSort(array, pivot_position + 1, high)
    end
```

```
end quickSort
```

```
algorithm partition(array <int>, val low <int>, val high <int>)  
    pivot = array[high]  
    position = low - 1  
    for (walker = low : high)  
        if (array[walker] <= pivot)  
            position ++  
            swap(array[position], array[walker])  
        end  
    end  
    return position + 1  
end
```

c Hiện thực: Hiện thực giải thuật bằng hợp ngữ MIPS

d Kết quả:

- Testcase 1:

- Input:

```
.data  
array: .word 15, 2, 3, 4, 10, 16, 17, 18, 19, 14  
        12, 13, 11, 8, 5, 6, 7, 20, 1, 9
```

- Output:

```
Quicksort: 15 2 3 4 10 16 17 18 19 14 12 13 11 8 5 6 7 20 1 9  
Quicksort: 2 3 4 8 5 6 7 1  
Quicksort: 3 4 8 5 6 7 2  
Quicksort: 4 8 5 6 7 3  
Quicksort: 8 5 6 7 4  
Quicksort: 5 6 7 8  
Quicksort: 5 6 7  
Quicksort: 5 6  
Quicksort: 9 14 12 13 11 15 10 16 17 20 18 19  
Quicksort: 9 14 12 13 11 15 10 16 17 18  
Quicksort: 9 14 12 13 11 15 10 16 17  
Quicksort: 9 14 12 13 11 15 10 16  
Quicksort: 9 14 12 13 11 15 10  
Quicksort: 12 13 11 15 14  
Quicksort: 12 13 11  
Quicksort: 13 12  
Quicksort: 13 14 15  
Quicksort: 13 14  
Quicksort: 19 20  
Sorted: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

- Testcase 2:

- Input:

```
.data  
array: .word 99, -22, 45, 12, -45, 126, 1007, 1218, 1339, -124  
        52, 4123, 121, -899, 5, 16, 37, 2, 1000, 92
```

- Output:

```
Quicksort: 99 -22 45 12 -45 126 1007 1218 1339 -124 52 4123 121 -899 5 16 37 2 1000 92
Quicksort: -22 45 12 -45 -124 52 -899 5 16 37 2
Quicksort: -22 -45 -124 -899
Quicksort: -45 -124 -22
Quicksort: -45 -124
Quicksort: -45 -22
Quicksort: 2 52 45 5 16 37 12
Quicksort: 2 5
Quicksort: 12 52 16 37 45
Quicksort: 12 16 37
Quicksort: 12 16
Quicksort: 45 52
Quicksort: 92 121 1007 1218 1339 99 126 1000 4123
Quicksort: 92 121 1007 1218 1339 99 126 1000
Quicksort: 92 121 99 126
Quicksort: 92 121 99
Quicksort: 121 126
Quicksort: 1000 1007 1218 1339
Quicksort: 1000 1007 1218
Quicksort: 1000 1007
Sorted: -899 -124 -45 -22 2 5 12 16 37 45 52 92 99 121 126 1000 1007 1218 1339 4123
```

- Testcase 3:

- Input:

```
.data
array: .word 0, 222, 865, -312, 4125, 6136, 507, -1518, -12119, -904
        72, 1223, 1331, 599, -523, 160, -327, 392, 10, -920
```

- Output:

```
Quicksort: 0 222 865 -312 4125 6136 507 -1518 -12119 -904 72 1223 1331 599 -523 160 -327 392 10 -920
Quicksort: -1518 -12119
Quicksort: -1518 -920 -312 4125 6136 507 0 222 -904 72 1223 1331 599 -523 160 -327 392 10 865
Quicksort: -1518 -920 -312 507 0 222 -904 72 599 -523 160 -327 392 10
Quicksort: -1518 -920 -312 0 -904 -523 -327
Quicksort: -1518 -920 -904 -523
Quicksort: -1518 -920 -904
Quicksort: -1518 -920
Quicksort: -327 0 -312
Quicksort: 0 10 599 222 160 507 392 72
Quicksort: 0 10
Quicksort: 72 222 160 507 392 599
Quicksort: 72 222 160 507 392
Quicksort: 72 222 160
Quicksort: 222 392 507
Quicksort: 222 392
Quicksort: 865 1331 6136 4125 1223
Quicksort: 6136 4125 1331
Quicksort: 4125 6136
Sorted: -12119 -1518 -920 -904 -523 -327 -312 0 10 72 160 222 392 507 599 865 1223 1331 4125 6136
```

- Testcase 4:

- Input:

```
.data
array: .word 120, 872, -365, 512, -4125, -636, 549, 120, -119, -704
        27, -523, -44631, 729, -943, 720, -69, 420, -10, 4000
```

- Output:

```
Quicksort: 120 872 -365 512 -4125 -636 549 120 -119 -704 27 -523 -44631 729 -943 720 -69 420 -10 4000
Quicksort: 120 872 -365 512 -4125 -636 549 120 -119 -704 27 -523 -44631 729 -943 720 -69 420 -10
Quicksort: -365 -4125 -636 -119 -704 -523 -44631 -943 -69
Quicksort: -365 -4125 -636 -119 -704 -523 -44631 -943
Quicksort: -4125 -44631
Quicksort: -4125 -943 -119 -704 -523 -365 -636
Quicksort: -4125 -943 -704
Quicksort: -4125 -943
Quicksort: -636 -523 -365 -119
Quicksort: -636 -523 -365
Quicksort: -636 -523
Quicksort: -10 27 120 549 729 120 720 512 420 872
Quicksort: -10 27 120 549 729 120 720 512 420
Quicksort: -10 27 120 120
Quicksort: -10 27
Quicksort: 120 120
Quicksort: 120 420 549 720 512 729
Quicksort: 120 420 549 720 512
Quicksort: 120 420
Quicksort: 512 720 549
Quicksort: 720 729
Sorted: -44631 -4125 -943 -704 -636 -523 -365 -119 -69 -10 27 120 120 420 512 549 720 729 872 4000
```

- Testcase 5:

- Input:

```
.data
array: .word 525, 772, -125, 312, -8125, -736, 4549, -120, -219, -219
        56, -53, 4151, 7, -219, 460, 99, -10, 100, 40000
```

- Output:

```
Quicksort: 525 772 -125 312 -8125 -736 4549 -120 -219 -219 56 -53 4151 7 -219 460 99 -10 100 40000
Quicksort: 525 772 -125 312 -8125 -736 4549 -120 -219 -219 56 -53 4151 7 -219 460 99 -10 100
Quicksort: -125 -8125 -736 -120 -219 -219 56 -53 7 -219 99 -10
Quicksort: -125 -8125 -736 -120 -219 -219 -53 -219
Quicksort: -8125 -736
Quicksort: -219 -120 -219 -219 -53 -125
Quicksort: -219 -219 -219
Quicksort: -219 -219
Quicksort: -219 -125 -53 -120
Quicksort: -219 -125
Quicksort: -120 -53
Quicksort: -10 56 99 7
Quicksort: 99 56
Quicksort: 99 100 772 525 460 4549 312 4151
Quicksort: 99 100 772 525 460 312
Quicksort: 99 100
Quicksort: 312 525 460 772
Quicksort: 312 525 460
Quicksort: 525 772
Quicksort: 4151 4549
Sorted: -8125 -736 -219 -219 -219 -125 -120 -53 -10 7 56 99 100 312 460 525 772 4151 4549 40000
```

- Testcase 6:

- Input:

```
.data
array: .word 5, 28, -523, 12, -4625, 183, 4928, -2305, 629, 21379
        56, -5328, 464, 28, -1296, 860, 9, 1053, -10038, 8000
```

- Output:

```
Quicksort: 5 28 -523 12 -4625 183 4928 -2305 629 21379 56 -5328 464 28 -1296 860 9 1053 -10038 8000
Quicksort: 5 28 -523 12 -4625 183 4928 -2305 629 56 -5328 464 28 -1296 860 9 1053 -10038
Quicksort: 28 -523 12 -4625 183 4928 -2305 629 56 -5328 464 28 -1296 860 9 1053 5
Quicksort: -523 -4625 -2305 -5328 -1296
Quicksort: -4625 -2305 -5328
Quicksort: -2305 -4625
Quicksort: -2305 -1296 -523
Quicksort: -2305 -1296
Quicksort: 5 12 629 56 28 464 28 183 860 9 1053 4928
Quicksort: 5 12 629 56 28 464 28 183 860 9 1053
Quicksort: 5 12 629 56 28 464 28 183 860 9
Quicksort: 629 56 28 464 28 183 860 12
Quicksort: 56 28 464 28 183 860 629
Quicksort: 56 28 464 28 183
Quicksort: 56 28 28
Quicksort: 28 56
Quicksort: 183 464
Quicksort: 629 860
Quicksort: 8000 21379
Sorted: -10038 -5328 -4625 -2305 -1296 -523 5 9 12 28 28 56 183 464 629 860 1053 4928 8000 21379
```

- Testcase 7:

- Input:

```
.data
array: .word 525, 328, 205, -206, 8364, 12973, -94928, 8095, -2929, 22217
        542, -60328, 424, 22158, -39296, 8180, 7285, -30293, 1038, 8020
```

- Output:

```
Quicksort: 525 328 205 -206 8364 12973 -94928 8095 -2929 22217 542 -60328 424 22158 -39296 8180 7285 -30293 1038 8020
Quicksort: 525 328 205 -206 -94928 -2929 542 -60328 424 -39296 7285 -30293 1038
Quicksort: 525 328 205 -206 -94928 -2929 542 -60328 424 -39296 -30293
Quicksort: -94928 -60328 -39296
Quicksort: -94928 -60328
Quicksort: -30293 525 -2929 542 328 424 205 -206
Quicksort: -30293 -2929
Quicksort: -206 542 328 424 205 525
Quicksort: -206 328 424 205
Quicksort: 424 328
Quicksort: 424 525 542
Quicksort: 424 525
Quicksort: 1038 7285
Quicksort: 8020 22217 8180 8364 8095 12973 22158
Quicksort: 8020 8180 8364 8095 12973
Quicksort: 8020 8180 8364 8095
Quicksort: 8364 8180
Quicksort: 8364 12973
Quicksort: 22158 22217
Sorted: -94928 -60328 -39296 -30293 -2929 -206 205 328 424 525 542 1038 7285 8020 8095 8180 8364 12973 22158 22217
```

• Testcase 8:

- Input:

```
.data
array: .word 55, 28, 2, 126, 64, 12, 98, 89, 929, 222
        512, 4428, 224, 158, 296, 90, 285, 329, 238, 20
```

- Output:

```
Quicksort: 55 28 2 126 64 12 98 89 929 222 512 4428 224 158 296 90 285 329 238 20
Quicksort: 2 12
Quicksort: 20 126 64 28 98 89 929 222 512 4428 224 158 296 90 285 329 238 55
Quicksort: 20 28
Quicksort: 55 126 98 89 929 222 512 4428 224 158 296 90 285 329 238 64
Quicksort: 98 89 929 222 512 4428 224 158 296 90 285 329 238 126
Quicksort: 98 89 90
Quicksort: 98 126 512 4428 224 158 296 929 285 329 238 222
Quicksort: 98 126 158
Quicksort: 98 126
Quicksort: 222 224 512 296 929 285 329 238 4428
Quicksort: 222 224 512 296 929 285 329 238
Quicksort: 222 224
Quicksort: 238 296 929 285 329 512
Quicksort: 238 296 285 329
Quicksort: 238 296 285
Quicksort: 296 329
Quicksort: 512 929
Sorted: 2 12 20 28 55 64 89 90 98 126 158 222 224 238 285 296 329 512 929 4428
```

• Testcase 9:

- Input:

```
.data
array: .word 27, -128, 232, 26, 24, 12, -28, 37, -259, 482
        51, 48, 2424, -1245, 196, 28, -2355, 39, -4838, 387
```

- Output:

```
Quicksort: 27 -128 232 26 24 12 -28 37 -259 482 51 48 2424 -1245 196 28 -2355 39 -4838 387
Quicksort: 27 -128 232 26 24 12 -28 37 -259 51 48 -1245 196 28 -2355 39 -4838
Quicksort: -128 232 26 24 12 -28 37 -259 51 48 -1245 196 28 -2355 39 27
Quicksort: -128 26 24 12 -28 -259 -1245 -2355
Quicksort: 26 24 12 -28 -259 -1245 -128
Quicksort: -259 -1245
Quicksort: -259 -128 -28 26 24 12
Quicksort: -259 -128 -28
Quicksort: -259 -128
Quicksort: 12 24 26
Quicksort: 12 24
Quicksort: 27 48 37 196 28 232 39 51
Quicksort: 27 48 37 28 39
Quicksort: 27 37 28
Quicksort: 37 39 48
Quicksort: 37 39
Quicksort: 51 196 232
Quicksort: 51 196
Quicksort: 387 2424 482
Sorted: -4838 -2355 -1245 -259 -128 -28 12 24 26 27 28 37 39 48 51 196 232 387 482 2424
```

- Testcase 10:

- Input:

```
.data
array: .word -295, -158, -152, 194, 895, 12, 28, 37, -359, 482
        112, 278, 417, -435, 196, 28, -575, 149, 3738, 7
```

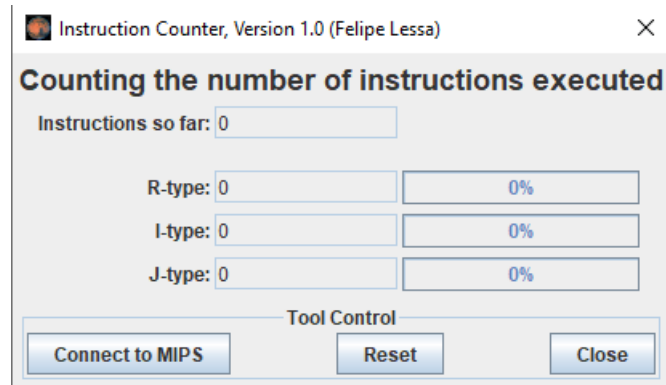
- Output:

```
Quicksort: -295 -158 -152 194 895 12 28 37 -359 482 112 278 417 -435 196 28 -575 149 3738 7
Quicksort: -295 -158 -152 -359 -435 -575
Quicksort: -158 -152 -359 -435 -295
Quicksort: -359 -435
Quicksort: -359 -295 -152 -158
Quicksort: -359 -295
Quicksort: -158 -152
Quicksort: 7 37 194 482 112 278 417 895 196 28 12 149 3738 28
Quicksort: 7 12
Quicksort: 28 482 112 278 417 895 196 28 37 149 3738 194
Quicksort: 28 112 28 37 149
Quicksort: 28 112 28 37
Quicksort: 28 28
Quicksort: 28 37 112
Quicksort: 28 37
Quicksort: 194 196 482 278 417 3738 895
Quicksort: 194 196 482 278 417
Quicksort: 194 196 278
Quicksort: 194 196
Quicksort: 417 482
Quicksort: 895 3738
Sorted: -575 -435 -359 -295 -158 -152 7 12 28 28 37 112 149 194 196 278 417 482 895 3738
```

4.2 Thống kê số lệnh, loại lệnh đã sử dụng

a Công cụ:

- Sử dụng công cụ Instruction Counter được cung cấp bởi MARS (Tool/Instruction Counter) xác định tổng số lệnh đã thực thi của chương trình.

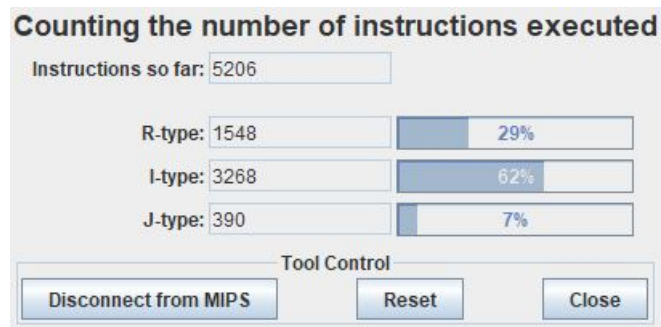


Trong đó:

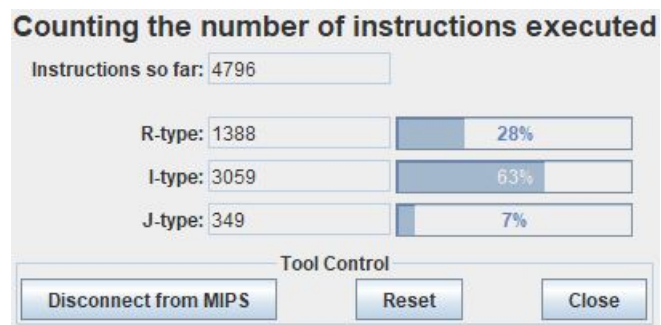
- Instructions so far là tổng số câu lệnh mà chương trình đã thực thi.
- R-type là tổng số câu lệnh loại R mà chương trình đã thực thi.
- I-type là tổng số câu lệnh loại I mà chương trình đã thực thi.
- J-type là tổng số câu lệnh loại J mà chương trình đã thực thi.

b Kết quả:

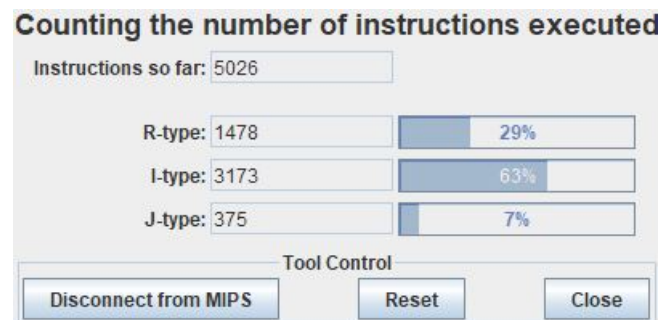
- Testcase 1:



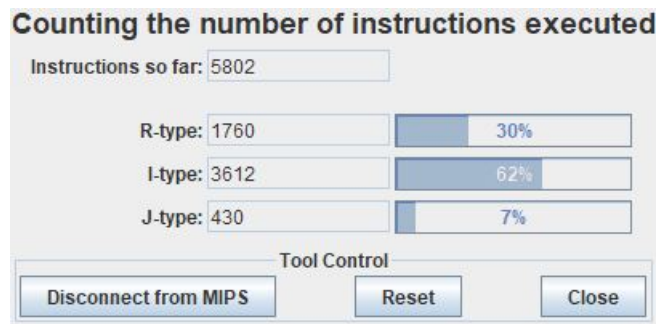
- Testcase 2:



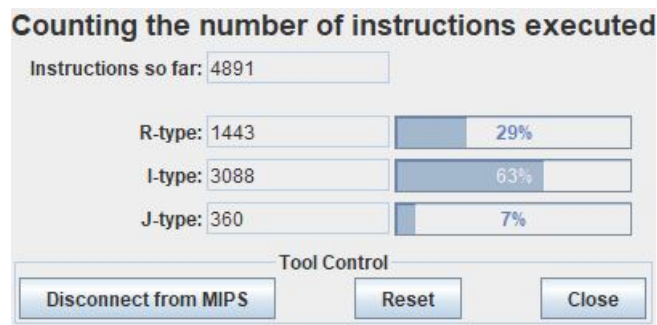
- Testcase 3:



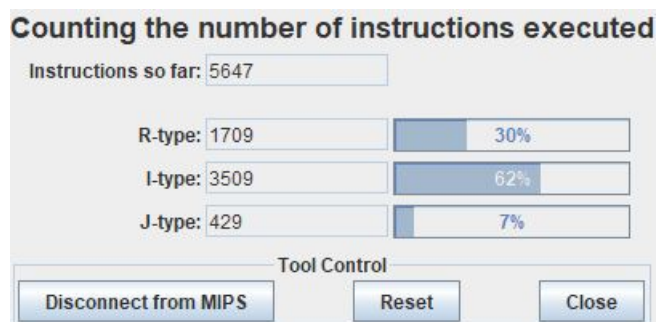
- Testcase 4:



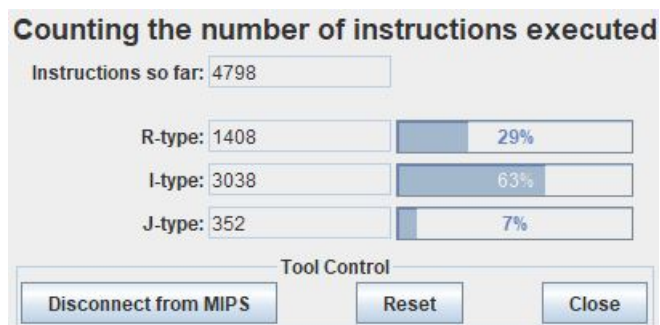
- Testcase 5:



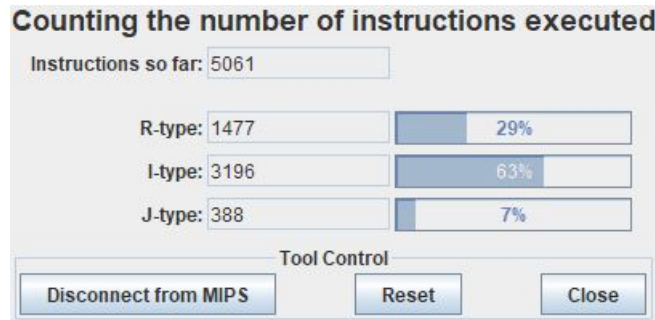
- Testcase 6:



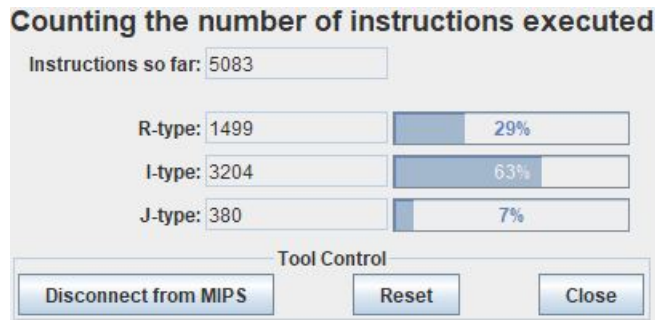
- Testcase 7:



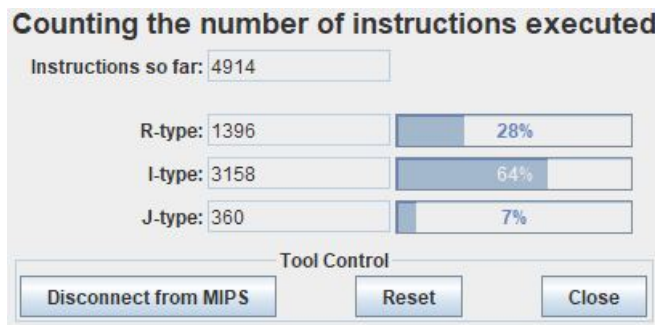
- Testcase 8:



- Testcase 9:



- Testcase 10:



4.3 Tính và trình bày cách tính thời gian chạy của chương trình

a Các bước thực hiện

- Xác định tổng số câu lệnh chương trình đã thực thi (đã thực hiện ở mục 3.2).
- Xác định thời gian thực thi của chương trình bằng công thức sau:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate}$$

Trong đó:

- CPU Execution Time là thời gian thực thi của CPU.
- Clock cycles là số chu kỳ xung nhịp cần để thực thi chương trình
- Instruction Count là tổng số câu lệnh chương trình đã thực thi
- CPI là số chu kỳ xung nhịp trên mỗi câu lệnh (trong trường hợp này giả sử CPI mỗi lệnh đều bằng 1).
- Clock rate là tần số của 1 xung nhịp (chúng ta sẽ sử dụng máy tính MIPS có tần số 2GHz).

b Tính toán:

- Testcase 1:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate} = \frac{5206}{2 \times 10^9} = 2.603 (\mu s)$$

- Testcase 2:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate} = \frac{4796}{2 \times 10^9} = 2.3975\ (\mu s)$$

- Testcase 3:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate} = \frac{5026}{2 \times 10^9} = 2.513\ (\mu s)$$

- Testcase 4:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate} = \frac{5802}{2 \times 10^9} = 2.901\ (\mu s)$$

- Testcase 5:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate} = \frac{4891}{2 \times 10^9} = 2.4455\ (\mu s)$$

- Testcase 6:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate} = \frac{5647}{2 \times 10^9} = 2.8235\ (\mu s)$$

- Testcase 7:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate} = \frac{4798}{2 \times 10^9} = 2.399\ (\mu s)$$

- Testcase 8:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate} = \frac{5061}{2 \times 10^9} = 2.5305\ (\mu s)$$

- Testcase 9:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate} = \frac{5083}{2 \times 10^9} = 2.5415\ (\mu s)$$

- Testcase 10:

$$CPUExecutionTime = \frac{Clock\ cycles}{Clock\ rate} = \frac{Instruction\ Count \times CPI}{Clock\ rate} = \frac{4914}{2 \times 10^9} = 2.457\ (\mu s)$$



5 Tài liệu tham khảo

- [1] Phạm Quốc Cường, Kiến trúc máy tính, NXB Đại học quốc gia Hồ Chí Minh
- [2] QuickSort. <https://www.geeksforgeeks.org/quick-sort/>.