

Computer Organization Lab-5

Members

Aiswarya H, 111901006

Joel Sam Mathew, 111901026

Reports

1. even-odd.asm

```
1      .data
2  ✓ a:
3      10
4      .text
5  ✓ main:
6      load %x0, $a, %x3
7      divi %x3, 2, %x4
8      muli %x4, 2, %x5
9      beq %x3, %x5, even
10     bne %x3, %x5, odd
11  ✓ even:
12     subi %x0, 1, %x10
13     end
14  ✓ odd:
15     addi %x0, 1, %x10
16     end
```

even-odd_stat.txt

```
1  Number of instructions executed = 6
2  Number of cycles taken = 23
3  Number of times OF got stalled = 6
4  Number of times wrong path was taken = 1
```

2. Fibonacci.asm

```
1      .data
2      n:
3      10
4      .text
5      main:
6          load %x0, $n, %x3
7          add %x0, %x0, %x4
8          addi %x0, 1, %x5
9          addi %x0, 0, %x6
10         addi %x0, 65535, %x9
11         store %x6, 0, %x9
12         subi %x9, 1, %x9
13         add %x0, %x5, %x6
14         store %x6, 0, %x9
15         addi %x0, 2, %x11
16     loop:
17         add %x4, %x5, %x6
18         add %x0, %x5, %x4
19         add %x0, %x6, %x5
20         subi %x9, 1, %x9
21         store %x6, 0, %x9
22         addi %x11, 1, %x11
23         beq %x11, %x3, endl
24         jmp loop
25     endl:
26     end
```

Fibonacci_stat.txt

```
1      Number of instructions executed = 74
2      Number of cycles taken = 179
3      Number of times OF got stalled = 27
4      Number of times wrong path was taken = 8
```

3. palindrome.asm

```
1      .data
2      a:
3          12321
4      .text
5      main:
6          load %x0, $a, %x3
7          load %x0, $a, %x5
8          add %x0, %x0, %x9
9          addi %x0, 1, %x4
10     loop:
11         blt %x5, %x4, endl
12         divi %x5, 10, %x6
13         muli %x6, 10, %x7
14         sub %x5, %x7, %x8
15         muli %x9, 10, %x9
16         add %x9, %x8, %x9
17         divi %x5, 10, %x5
18         jmp loop
19     endl:
20         beq %x9, %x3, success
21         bne %x9, %x3, fail
22     success:
23         addi %x0, 1, %x10
24         end
25     fail:
26         subi %x0, 1, %x10
27         end
```

palindrome_stat.txt

```
1      Number of instructions executed = 48
2      Number of cycles taken = 119
3      Number of times OF got stalled = 22
4      Number of times wrong path was taken = 7
```

4. prime.asm

```
1      .data
2      a:
3          17
4      .text
5      main:
6          load %x0, $a, %x3
7          addi %x0, 2, %x6
8      loop:
9          div %x3, %x6, %x5
10         mul %x5, %x6, %x7
11         beq %x3, %x7, fail
12         addi %x6, 1, %x6
13         beq %x6, %x3, endl
14         jmp loop
15     fail:
16         subi %x0, 1, %x10
17         end
18     endl:
19         addi %x0, 1, %x10
20         end
```

prime_stat.txt

```
1      Number of instructions executed = 93
2      Number of cycles taken = 256
3      Number of times OF got stalled = 77
4      Number of times wrong path was taken = 15
```

5. descending.asm

```
1      .data
2      a:
3          70
4          80
5          40
6          20
7          10
8          30
9          50
10         60
11     n:
12         8
13     .text
14     main:
15         load %x0, $a, %x3
16         subi %x0, 1, %x10
17         load %x0, $n, %x4
18     loopi:
19         addi %x10, 1, %x10
20         subi %x0, 1, %x11
21         beq %x10, %x4, finish
22     loopj:
23         addi %x11, 1, %x11
24         sub %x4, %x10, %x9
25         subi %x9, 1, %x13
26         addi %x11, 1, %x12
27         beq %x11, %x13, endl
28         load %x11, $a, %x5
29         load %x12, $a, %x6
30         blt %x5, %x6, swap
31         jmp loopj
32     endl:
33         jmp loopi
34     swap:
35         store %x6, $a, %x11
36         store %x5, $a, %x12
37         jmp loopj
38     finish:
39         end
40
```

descending_stat.txt

```
1      Number of instructions executed = 355
2      Number of cycles taken = 764
3      Number of times OF got stalled = 109
4      Number of times wrong path was taken = 57
```