Project 1 Report
Name: Fanyue Zhu Student id: 522031910547

---

1 Basic

The the updated sample program and its output are shown in the figures.

```
# file: example.asm

.data
string1: .asciiz "Student ID: 522031910547\n"
string2: .asciiz "Name: Fanyue Zhu\n"
.text
main:

li $v0, 4
la $a0, string1
syscall

li $v0, 4
la $a0, string2
syscall

li $v0, 10
syscall
```

Figure 1: modified sample program

```
Student ID: 522031910547
Name: Fanyue Zhu


-- program is finished running --
```

Figure 2: Output of modified sample program

2 Task: Branch

2.1 Coding

The assembly code is shown below.

```
1   #file: branch.asm
2   .data
3   A: .word 10
4   B: .word 15
5   C: .word 6
6   Z: .word 0
7   .text
8   main:
9   lw $t5, A
10  lw $t6, B
11  lw $t7, C
12  lw $t8, Z
13  bgt $t5, $t6, L10
14  blt $t7, 5, L10
15  j L110
16  L10:
17  li $t8, 1
18  j L20
19  L110:
20  beq $t5, $t6, L11
21  li $t8, 3
22  j L20
23  L11:
24  li $t8, 2
25  L20:
26  beq $t8, 1, L3
27  j L4
28  L3:
29  li $t8, -1
30  j L7
31  L4:
32  beq $t8, 2, L5
33  j L6
34  L5:
35  li $t8, -2
36  j L7
37  L6:
38  li $t8, 0
39  L7:
40  sw $t8, Z
41
```

Figure 3: Code of branch task

2.2 Result

The running result of all cases are shown as below.

(a) Case1: A=10 B=15 C=1

(b) Case2: A=15 B=10 C=6

(c) Case3: A=15 B=15 C=6

(d) Case4: A=10 B=15 C=6

Figure 4: Result of branch task

2.3   In-depth Investigation

As the figures shown below, in **slt $t6 at $t5**, PC, Instruction Memory, Registers(read port and write port), ALU and PC+4 adder are used. And in **bne $zero at 3**, since this branch statement satisfies the jump condition, PC, Instruction Memory, Registers(read and write port), ALU, PC+4 adder and Branch adder are used.
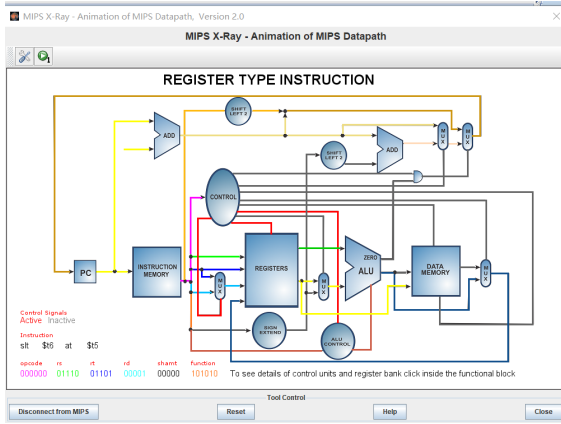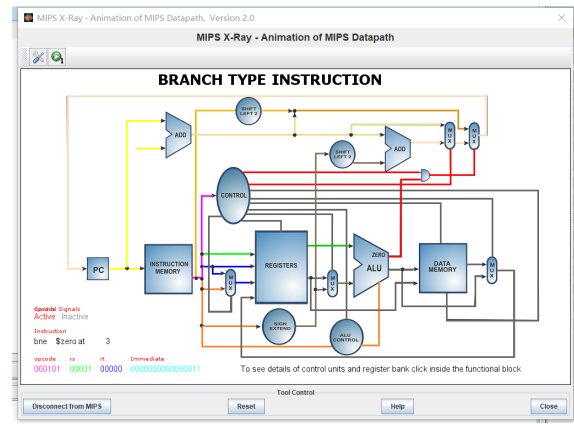

(a) **slt $t6 at $t5**

(b) **bne $zero at 3**

Figure 5: MIP X-Ray

3 Task: Loop

3.1 Coding

The assembly code is shown as below.

```
1   #file:loop.asm
2   .data
3   Z:.word 99
4   i:.word 0
5   .text
6   main:
7   lw $t1, Z
8   lw $t2, i
9   Loop1:
10  add $t2, $t2, 1
11  blt $t2, $t1, Loop1
12
13  bgt $t2, 0, Loop2
14  Loop2:
15  add $t1, $t2, $t1
16  add $t2, $t2, -1
17  bgt $t2, 0, Loop2
18
19  sw $t1, Z
20  sw $t2, i
21
```

Figure 6: Code of loop task

3.2   Result

The running result of both cases are shown as below.

(a) Case1: Z=100 i=0

(b) Case2: Z=99 i=0

Figure 7: Result of branch task

3.3   Optimization

Simplify the number of calculations by unrolling the loop into groups of 16 and recording 16*i with a new variable t for each group. Finally, the last group that is less than 16 is solved by traversing it. This reduces the number of calculations for each case to less than 400.

The numbers of instructions required for program execution are shown as below.

(a) Case1: Z=100 i=0

(b) Case2: Z=99 i=0

Figure 8: Optimization for the number of instructions