

# HW1\_Report

林炫廷 111550087

July 14, 2023

## 1. What is the usage of \$zero? What happen if you execute?

addi \$zero, \$zero, 5

**Ans:**

*First.*

1. This register could be a constant zero used in arithmetic and logical operation. Also, it is admitted to be considered as value zero in the instructions.
2. It could be used for register initialization, making it easy to clear the variables and registers.
3. It is commonly used in comparisons and branching. For example: *bne* and *beq*.

*Second.*

The register \$zero reserves the value zero. Unlike other general-purpose registers, you cannot modify the value of \$zero directly. Any attempt to write to \$zero is ignored, and it remains set to zero. As a result, this instruction does not affect register \$zero. and it retains its value of zero.

**2. How to use the stack to ensure that the value of each register is correctly saved when executing a recursive function?**

**Ans:**

After entering a function, we simply use stack to store the return address and other parameters. This allows us to obtain the correct parameters within the function layer and return to the correct line of code. The table below illustrates this process:

addi \$sp, \$sp, -8
sw \$ra, 4(\$sp)
sw \$a0, 0(\$sp)

By utilizing the stack, we maintain a hierarchical structure of function calls, ensuring the integrity of parameter passing and return addresses throughout the program execution.

**3. What was the most challenging part for you in this homework?**

**Ans:**

There are lots of obstacles that I have encountered while working on this homework assignment. The main challenge I face is that MIPS code is quite abstract, making it difficult to determine whether the program is progressing correctly. To implement the for loop function in MIPS, I have searched through the class slide and gathering information from Internet.

It is obvious that debugging the code is more challenging compared to high-level languages. This is particularly true when doing recursive functions. Despite carefully reading each line of code, I still struggle to identify why the program's output is consistently incorrect. Thus, I have resorted to closely observing the change in registers after executing each instruction. By employing this approach, I have been able to identify bugs within the complex program structure.

Although it is not easy to solve these question by using the unfamiliar language, It has significantly accelerated my familiarity with assembly language. I believe this homework is truly valuable in enhancing my understanding of this course.