Arithmetic Operations

5 min

Arithmetic operations make up the majority of functions in the Assembly languages.

After all, the manipulation of

Preview: Docs Binary is a number system of 1s and 0s which serves as a textual representation method for giving instructions to a computers CPU. In computing and telecommunications, binary codes are used for various methods of encoding data such as turning the character strings of source code, into bit strings or instruction sets for the CPU to execute.

binary

numbers is how you execute any type of code, whether that be changing a character from lowercase to upper case or turning a pixel on a screen from red to blue.

One of the most important aspects of mathematical operations is how numbers are stored at the machine hardware level. Memory locations, such as registers,

Preview: Docs Loading link description

cache

, or secondary storage, all have fixed binary lengths.

These fixed lengths mean that processors must have special registers to “catch” overflow from operations. Overflow operations can include handling extremely large numbers, marking when a carryover occurs in addition or storing both the quotient and remainder of division operations.

Most languages allow for at least two distinct types of each arithmetic operation. One type performs a calculation on two registers and the other performs an operation using one value from a register and another value that is directly added, known as an immediate or constant.

The addition calculation for example can be called using the ADD opcode which then takes three registers as operands, two registers to add together and another to store the answer in.

The addition function also has an ADDI command, which takes a register address and a constant to operate on and a register to store the answer in.

In the ADD function, the value in $5 is added to the value in $4 and stored in $6.

ADD $4, $5, $6

to Clipboard

In the ADDI function, the constant 7 is added to the value in $4 and stored in $6

ADDI $4, $6, 7

to Clipboard

Other common arithmetic operations include: SUB, SUBI, MULT, MULTI, DIV, and DIVI.

Instructions

1. Checkpoint 1 Passed

1.

Create a new variable, answer_1, and set it equal to the number of registers that are used in a typical ADD statement.

Hint

Remember we need things to operate on and a place to store the answer.

2. Checkpoint 2 Passed

2.

Create another variable, answer_2, and set it equal to the value of the constant in this statement: ADDI ($5), $9, 15

Hint

Reference the ADDI statement in the exercise to get the format of the function. A constant in Assembly is another name for an Immediate value.

arithmetic_operations.py

```
# Write your answers here:

answer_1 = 3

answer_2 = 15
```