Montgomery Yu Computer Organization Assignment 4: MIPS#1

RAT:

$t0 = X

$t1 = Y

$t2 = Z

MIPS Code (IfElse.asm):

**.data**

**x: .word 3 # assign value to x – will be initialized as instructed**

**y: .word 3 # assign value to y – will be initialized as instructed**

**z: .word 0 # assign value 0 to z**

**xResult: .asciiz "X=" # declare string variable for printing x**

**yResult: .asciiz " Y=" # declare string variable for printing y**

**zResult: .asciiz " Z=" # declare string variable for printing z**

**.text**

**lw $t0, x # load value in x to $t0 in the register table**

**lw $t1, y # load value in y to $t1 in the register table**

**bne $t0, $t1, else1 # If the value in $t0 is NOT equal to the value in $t0 -> if yes go to else1 section, otherwise normally proceed the code**

**add $t2, $t0, $t1 # Add $t0 and $t1, and storing that result into $t2**

**j printArea # After running the above code, jump to print Area**

**else1:**

**addi $t3, $zero, 3 # else1 section: adding an integer 3 to new location $t3 in the register table for later subtraction**

**sub $t3, $t0, $t3 # subtract 3 from $t0, and store the result in $t3**

**bne $t3, $t1, else2 # If the new result in $t3 is not equal to value in $t1, go to else2, otherwise proceed below as follows**

**lw $t2, z # Load the value 0 in the z variable to $t2**

**j printArea # After running this section, jump to print Area**

**else2:**

**sub $t2, $t0, $t1 # else2 section: subtract the value in $t1 from $t0, and store that result into $t2**

**j printArea # jump to printArea**

**printArea:**

**li $v0, 4 # put a 4 into $v0 - we want to print a string**

**la $a0, xResult # load String to display X**

**syscall # does whatever $v0 indicate**

**lw $a0, x # loading the value in x to the $a0 for printing**

**li $v0, 1 # put a 1 into $v0 - indicate we want to print a integer**

**syscall**

**li $v0, 4 # print a string**

**la $a0, yResult # loading string for displaying Y**

**syscall # execute**

**lw $a0, y # loading value in y to the $a0 for printing**

**li $v0, 1 # print integer**

**syscall**

**li $v0, 4 # printing a string**

**la $a0, zResult # loading string for displaying Z**

**syscall**

**move $a0, $t2 # move the current value store in $t2 to $a0 for printing**

**li $v0, 1 # print integer**

**syscall**

**exit:**

**li $v0, 10 # polite exit**

**syscall**

OUTPUT SCREENSHOT:

MIPS Code was run 3 times as instructed

Graphical user interface, text, application

Description automatically generated