##Overview

The implementation supports multiple cycle per instruction add, sub, lw, sw, beq and slt.

Based on the implementation scheme from chapter 5, The Processor: Datapath and Control (5.5 A multicycle implementation) of Computer

Organization and Design by David A. Patterson and John L. Hennessy, 3rd edition.

The memory is structured in 32-bit words.

Branches are finished in 3 cycles, R-type (ALU) instructions in 4, Stores (sw) in 4, Loads (lw) in 5.

There is only one memory which contains both data and instruction memory.

The starting address of the instruction memory is PC\_START = 128.

```

add $t0, $zero, $zero

add $t6, $zero, $zero

lw $t1, 64($t0)

lw $t2, 68($t0)

lw $t3, 72($t0)

sw $zero, 76($t0) #the sum will be at this location [76]

loop:

lw $t4, 0($t0)

lw $t5, 76($t6)

add $t5, $t5, $t4

sw $t5, 76($t6)

sub $t1, $t1, $t2

add $t0, $t0, $t3

beq $t1, $zero, done

beq $t1, $t1, loop #actually jump (because $t1 = $t1)

done:

#end

```

The program computes the sum of the first 16 values from the data memory.

The result will be 5 and will be located in the data memory.

#Tools

Modelsim was used for simulation. There is a free student edition available.

QtSpim was used to view the codes for each instruction.