ASSEMBLY CODE

# Load the value of 1 and 2 from memory into registers $t1 and $t2 respectively

lw $t1, 1021

lw $t2, 1022

VERILOG OUTPUT

# time 45 $v1: 0 $t1: 1 $t2: 0 $t3: 0 $t4: 0 $t5: 0 $t6: 0 $t7: 0 $s0: 0 $s1: 0 $sp: 0 $ra: 0

# time 95 $v1: 0 $t1: 1 $t2: 2 $t3: 0 $t4: 0 $t5: 0 $t6: 0 $t7: 0 $s0: 0 $s1: 0 $sp: 0 $ra: 0

ASSEMBLY CODE

# xor the value stored in $t1 (1) with 99

# expected output: 98

xori $t3, $t1, 99

VERILOG OUTPUT

# time 175 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 0 $t5: 0 $t6: 0 $t7: 0 $s0: 0 $s1: 0 $sp: 0 $ra: 0

ASSEMBLY CODE

# $t1 + $t2 = 1 + 2 = 3, stored in $t4

add $t4, $t1, $t2

VERILOG OUTPUT

# time 215 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 0 $t6: 0 $t7: 0 $s0: 0 $s1: 0 $sp: 0 $ra: 0

ASSEMBLY CODE

# $t2 - $t1 = 2 - 1 = 1, stored in $t5

sub $t5, $t2, $t1

VERILOG OUTPUT

# time 255 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 0 $t7: 0 $s0: 0 $s1: 0 $sp: 0 $ra: 0

ASSEMBLY CODE

# ($t1 < $t2) = (2 < 1) = 1, stored into $t6

slt $t6, $t1, $t2

VERILOG OUTPUT

# time 295 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 0 $s0: 0 $s1: 0 $sp: 0 $ra: 0

ASSEMBLY CODE

# if $t1 != $t5 (1 != 1) go to the jumpelseif

# because (1 != 1) is false, we'll do the $t1 + $t2 = 1 + 2 = 3, stored in $t7

# then we'll jump to the end

bne $t1, $t5, jumpelseif

add $t7, $t1, $t2

j jumpend

jumpelseif:

sub $t8, $t2, $t1

jumpend:

VERILOG OUTPUT

# time 365 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 0 $s1: 0 $sp: 0 $ra: 0

ASSEMBLY CODE

# A test of the difference function

# difference(8, 4) = 4

li $s0, 8 # first input

li $s1, 4 # second input

li $sp, 1020 # set the stack pointer to a specific location in memory

VERILOG OUTPUT

# time 405 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 8 $s1: 0 $sp: 0 $ra: 0

# time 425 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 8 $s1: 4 $sp: 0 $ra: 0

# time 445 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 8 $s1: 4 $sp: 1020 $ra: 0

ASSEMBLY CODE

jal difference

li $v0, 10

syscall

# FUNCTION: difference(a, b)

# Returns (a - b)

# Assumption: a > b

#

# Executes recursively (below is the psudocode)

# function difference (a, b)

# if a == b

# return 0

# else

# return 1 + difference(a - 1 , b)

difference:

bne $s0, $s1, end

# return 0

add $v1, $zero, $zero

jr $ra

end:

# pushing to the stack

sub $sp, $sp, $t4

sw $ra, 2($sp)

sw $s1, 1($sp)

sw $s0 0($sp)

# Argument prep (a - 1)

sub $s0, $s0, $t1

# jal

jal difference

# popping from the stack

lw $ra, 2($sp)

lw $s1, 1($sp)

lw $s0, 0($sp)

add $sp, $sp, $t4

# return 1 + difference($t1 , $t2)

add $v1, $v1, $t1

jr $ra

VERILOG OUTPUT

# time 465 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 8 $s1: 4 $sp: 1020 $ra: 15

# time 545 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 8 $s1: 4 $sp: 1017 $ra: 15

# time 705 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 7 $s1: 4 $sp: 1017 $ra: 15

# time 725 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 7 $s1: 4 $sp: 1017 $ra: 26

# time 805 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 7 $s1: 4 $sp: 1014 $ra: 26

# time 965 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 6 $s1: 4 $sp: 1014 $ra: 26

# time 1065 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 6 $s1: 4 $sp: 1011 $ra: 26

# time 1225 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 5 $s1: 4 $sp: 1011 $ra: 26

# time 1325 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 5 $s1: 4 $sp: 1008 $ra: 26

# time 1485 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 4 $s1: 4 $sp: 1008 $ra: 26

# time 1765 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 5 $s1: 4 $sp: 1008 $ra: 26

# time 1805 $v1: 0 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 5 $s1: 4 $sp: 1011 $ra: 26

# time 1845 $v1: 1 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 5 $s1: 4 $sp: 1011 $ra: 26

# time 2025 $v1: 1 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 6 $s1: 4 $sp: 1011 $ra: 26

# time 2065 $v1: 1 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 6 $s1: 4 $sp: 1014 $ra: 26

# time 2105 $v1: 2 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 6 $s1: 4 $sp: 1014 $ra: 26

# time 2285 $v1: 2 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 7 $s1: 4 $sp: 1014 $ra: 26

# time 2325 $v1: 2 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 7 $s1: 4 $sp: 1017 $ra: 26

# time 2365 $v1: 3 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 7 $s1: 4 $sp: 1017 $ra: 26

# time 2445 $v1: 3 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 7 $s1: 4 $sp: 1017 $ra: 15

# time 2545 $v1: 3 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 8 $s1: 4 $sp: 1017 $ra: 15

# time 2585 $v1: 3 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 8 $s1: 4 $sp: 1020 $ra: 15

# time 2625 $v1: 4 $t1: 1 $t2: 2 $t3: 98 $t4: 3 $t5: 1 $t6: 1 $t7: 3 $s0: 8 $s1: 4 $sp: 1020 $ra: 15