Week 2

1. Write an Assembly Program for:
2. addition of N words

.data

a: .word 0x00000023,0xffff456A

.text

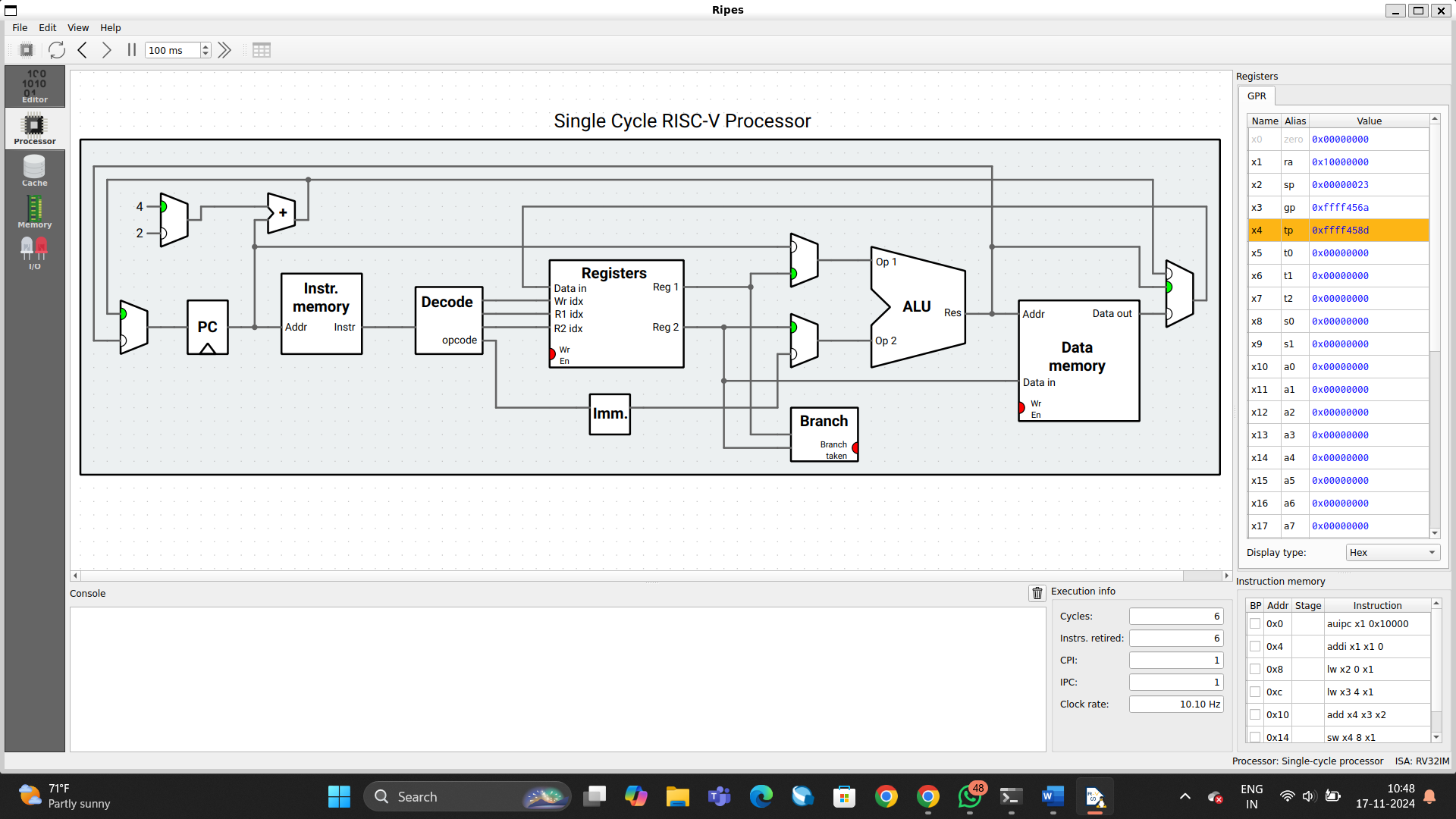
la x1,a

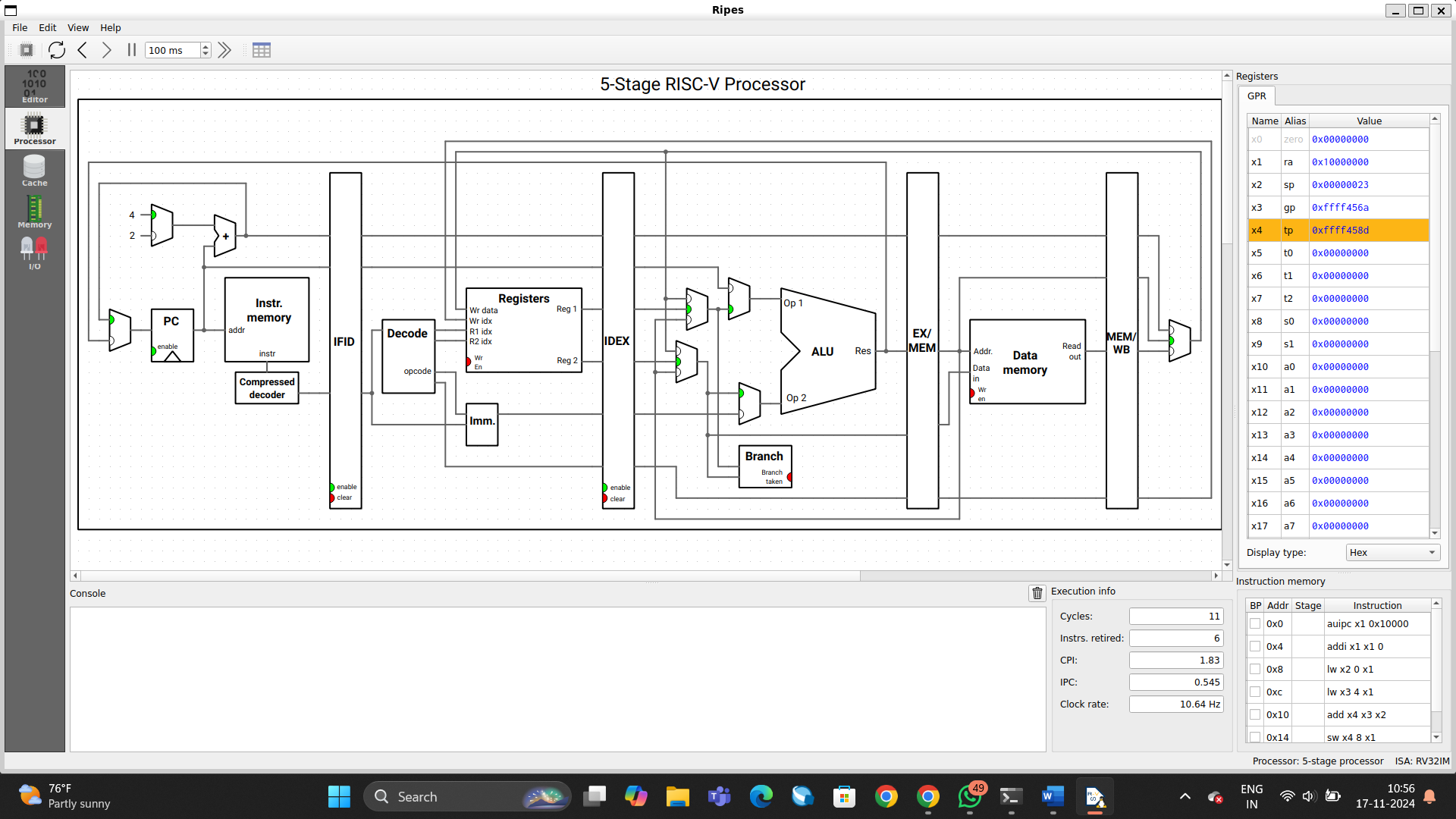
lw x2,0(x1)

lw x3,4(x1)

add x4,x3,x2

sw x4,8(x1)





1. addition of N half words

.data

a: .half 0x0001, 0x0089

.text

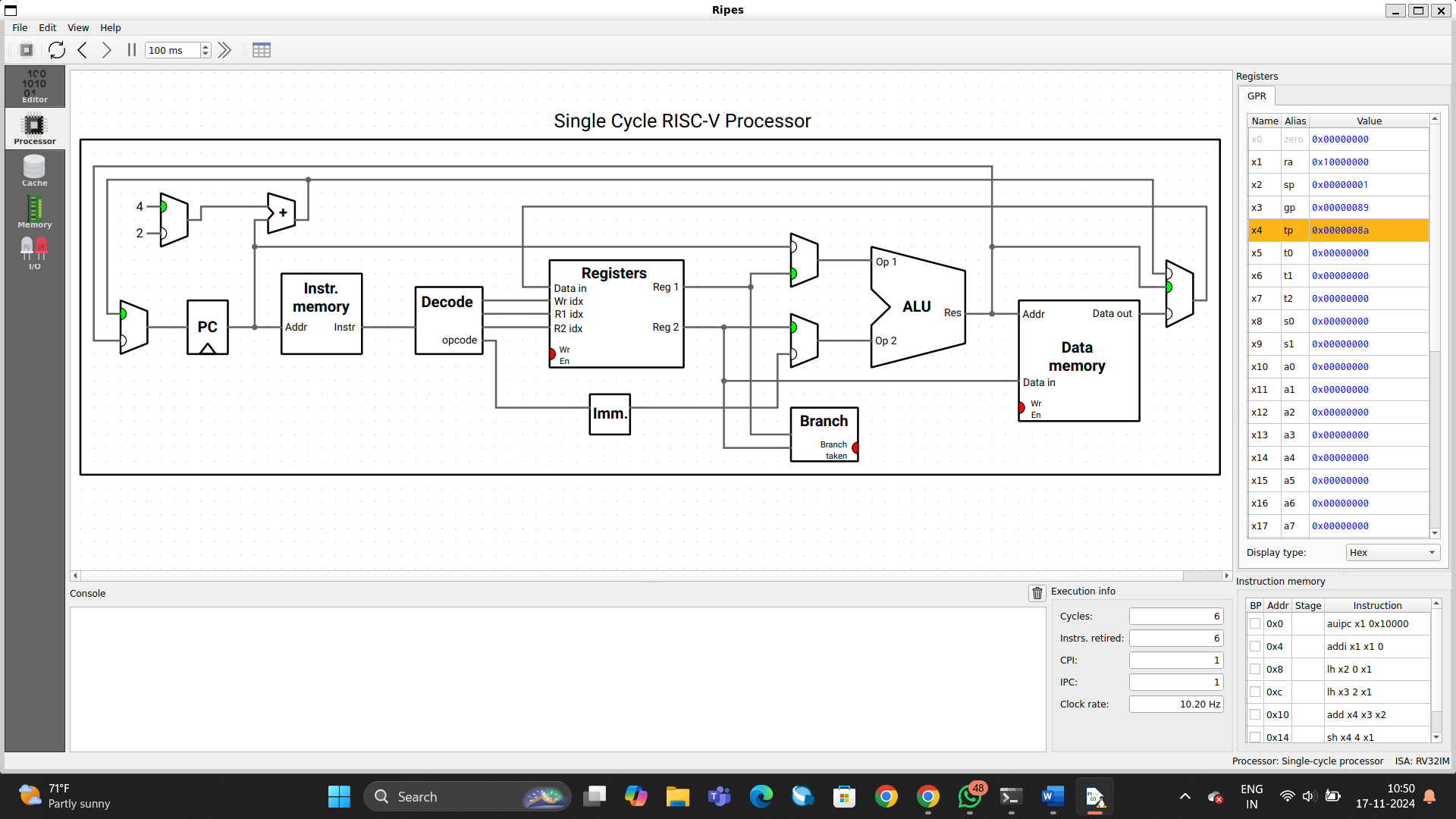
la x1, a

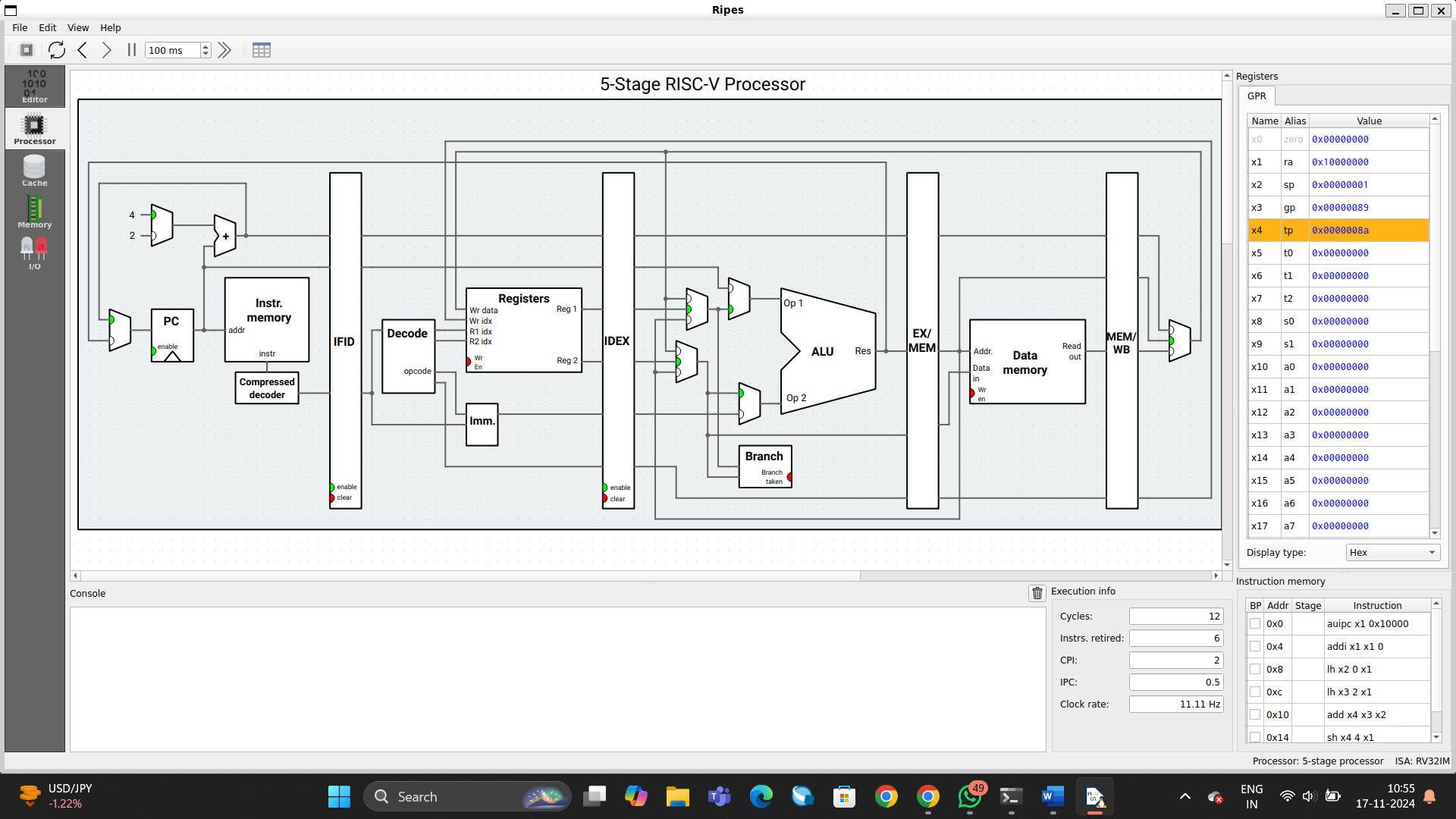
lh x2, 0(x1)

lh x3, 2(x1)

add x4, x3, x2

sh x4, 4(x1)





1. addition of N bytes

.data

a: .byte 0x02, 0x6A

.text

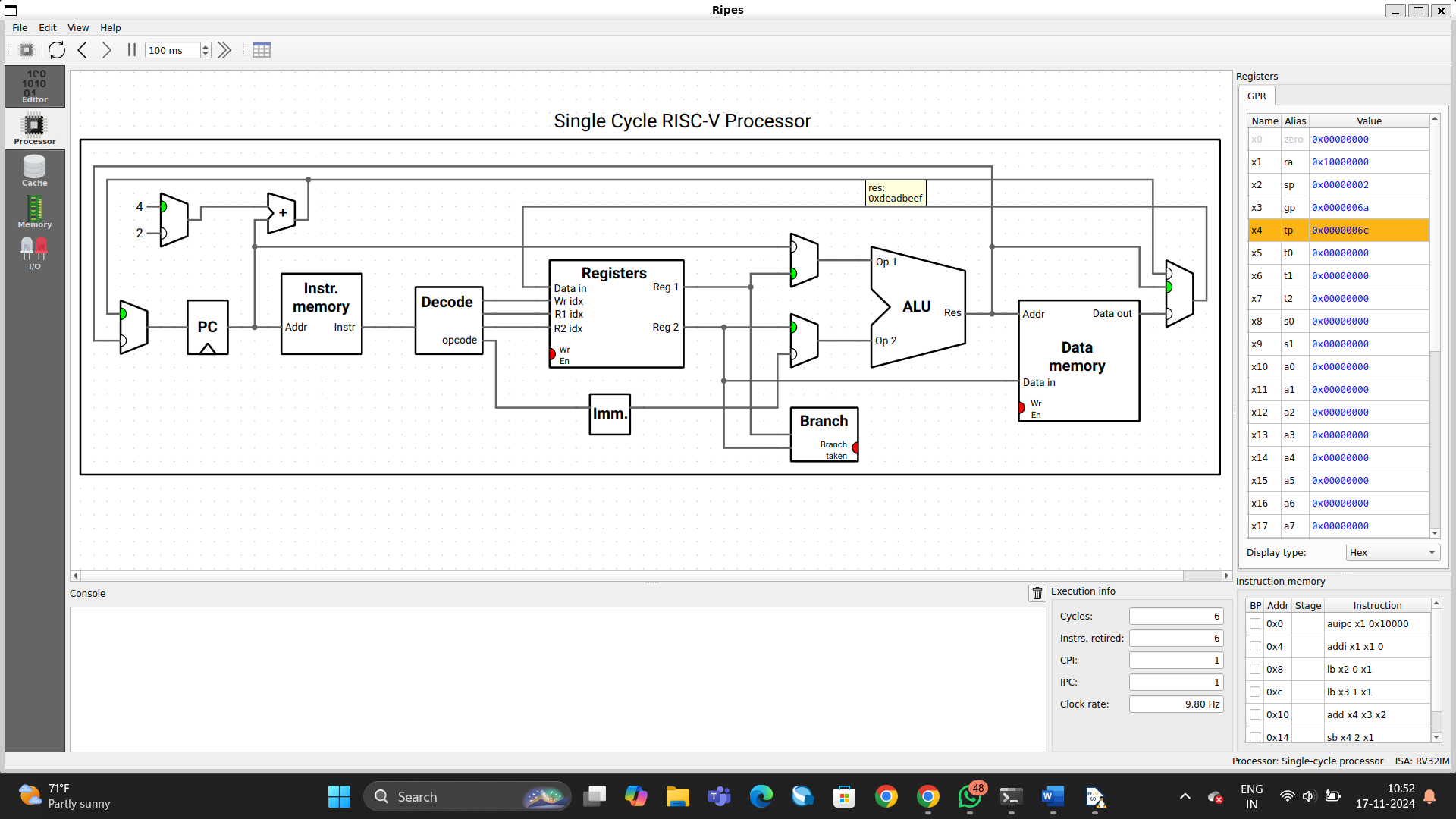
la x1, a

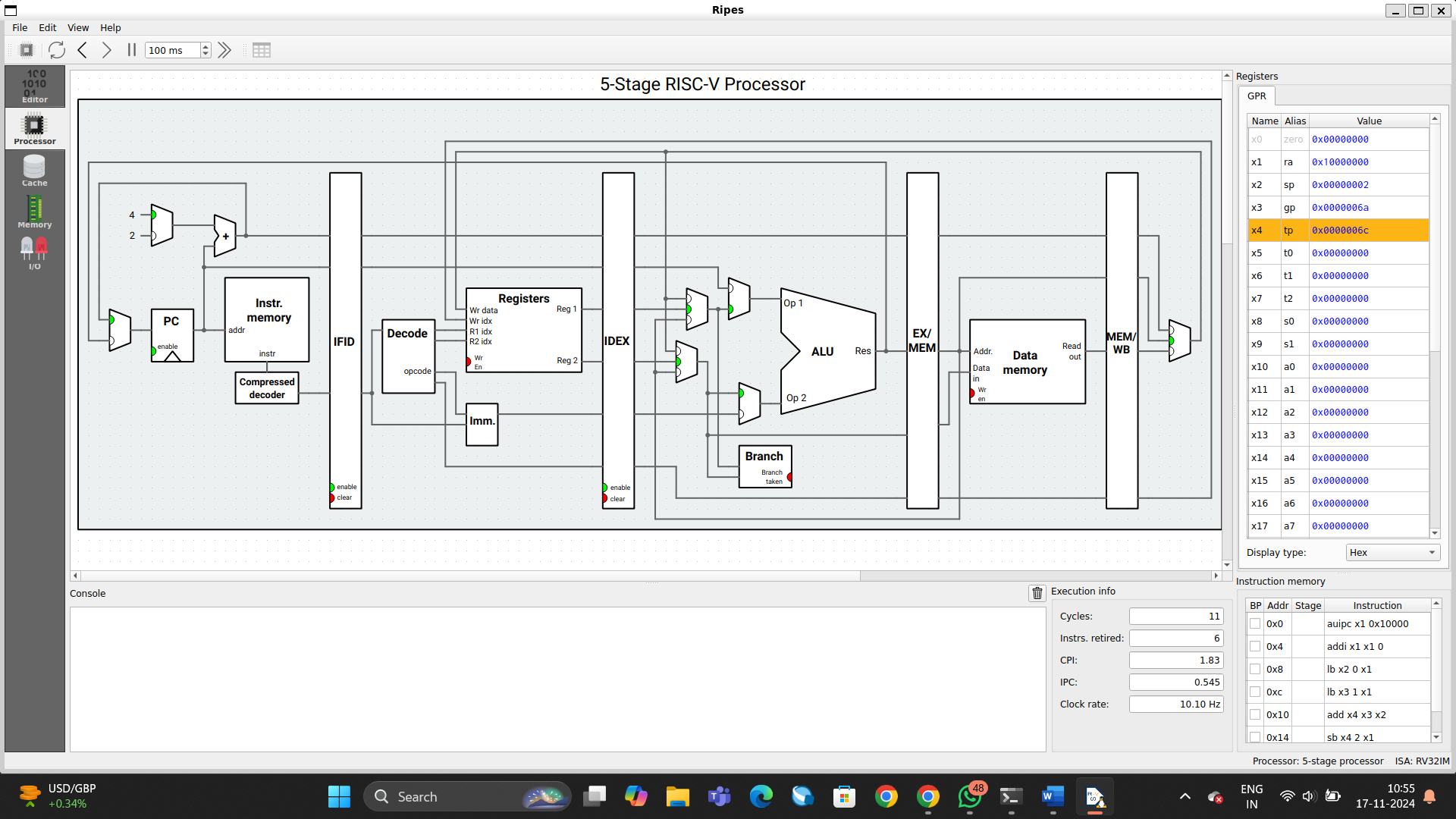
lb x2, 0(x1)

lb x3, 1(x1)

add x4, x3, x2

sb x4, 2(x1)





1. Write an Assembly program for calculating x = (y + m) - (L - D) + (Z + C) - D, where x, y, m, L, D, Z, C are elements of 32-bits wide

.data

y: .word 10 # Value of y

m: .word 20 # Value of m

L: .word 30 # Value of L

D: .word 5 # Value of D

Z: .word 50 # Value of Z

C: .word 40 # Value of C

x: .word 0 # To store the result (x)

.text

# Load values into registers

la x1, y # Load address of y

lw x2, 0(x1) # Load value of y into x2

la x3, m # Load address of m

lw x4, 0(x3) # Load value of m into x4

# Calculate y + m

add x5, x2, x4 # x5 = y + m

la x6, L # Load address of L

lw x7, 0(x6) # Load value of L into x7

la x8, D # Load address of D

lw x9, 0(x8) # Load value of D into x9

# Calculate L - D

sub x10, x7, x9 # x10 = L - D

# Calculate (y + m) - (L - D)

sub x11, x5, x10 # x11 = (y + m) - (L - D)

la x12, Z # Load address of Z

lw x13, 0(x12) # Load value of Z into x13

la x14, C # Load address of C

lw x15, 0(x14) # Load value of C into x15

# Calculate Z + C

add x16, x13, x15 # x16 = Z + C

# Calculate ((y + m) - (L - D)) + (Z + C)

add x17, x11, x16 # x17 = ((y + m) - (L - D)) + (Z + C)

# Calculate ((y + m) - (L - D)) + (Z + C) - D

sub x18, x17, x9 # x18 = ((y + m) - (L - D)) + (Z + C) - D

# Store result in x

la x19, x # Load address of x

sw x18, 0(x19) # Store the result of the calculation in x

# Exit the program (system call for exit)

li a7, 10 # Exit system call

ecall

