

```
#####
# MIPS Helper Full Instruction Test (with PC + Display Comments)
#####

.data
buffer: .space 32

.text
.globl main
main:

#####
# Arithmetic Instructions
#####

# PC = 0
addi $t0, $zero, 5      # t0 = 5, display 0, 5
nop
nop
nop
nop
nop
nop

# PC = 24
addi $t1, $zero, 3      # t1 = 3, display 24, 3
nop
nop
nop
nop
nop
nop

# PC = 48
add $t2, $t0, $t1      # t2 = 5 + 3 = 8, display 48, 8
nop
nop
nop
nop
nop
nop

# PC = 72
sub $t3, $t0, $t1      # t3 = 5 - 3 = 2, display 72, 2
nop
nop
nop
```

```
nop
nop

# PC = 96
mul $t4, $t0, $t1      # t4 = 5 * 3 = 15, display 96, 15
nop
nop
nop
nop
nop
nop

#####
# Data Memory Instructions
#####

# PC = 120
la $t5, buffer      # t5 = address of buffer, display 120, 0x10010000
nop
nop
nop
nop
nop
nop

# PC = 144
# SW rt, offset(rs)
# memory[rs] = t0
# memory[rs] = 5
sw $t0, 0($t5)      # memory[buffer+0] = 5, display 144, 0
nop
nop
nop
nop
nop

# PC = 168
# LW rt, offset(rs)
# rt = rs
# LW
lw $s0, 0($t5)      # s0 = memory[buffer+0] = 5, display 168, 5
nop
nop
nop
nop
nop
```

```
# PC = 192
    sb $t1, 4($t5)      # memory[buffer+4] = 3, display 192, 0
    nop
    nop
    nop
    nop
    nop
```

```
# PC = 216
    lb $s1, 4($t5)      # s1 = 3, display 216, 3
    nop
    nop
    nop
    nop
    nop
```

```
# PC = 240
    sh $t2, 6($t5)      # memory[buffer+6] = 8, display 240, 0
    nop
    nop
    nop
    nop
    nop
```

```
# PC = 264
    lh $s2, 6($t5)      # s2 = 8, display 264, 8
    nop
    nop
    nop
    nop
    nop
```

```
#####
# Logical Instructions
#####
```

```
# PC = 288
    and $t6, $t0, $t1     # t6 = 5 & 3 = 1, display 288, 1
    nop
    nop
    nop
    nop
    nop
```

```
# PC = 312
or $t7, $t0, $t1      # t7 = 5 | 3 = 7, display 312, 7
nop
nop
nop
nop
nop
nop

# PC = 336
nor $t8, $t0, $t1     # t8 = ~(5 | 3) = -8, display 336, -8
nop
nop
nop
nop
nop
nop

# PC = 360
xor $t9, $t0, $t1     # t9 = 5 ^ 3 = 6, display 360, 6
nop
nop
nop
nop
nop
nop

# PC = 384
andi $t6, $t6, 7       # t6 = 1 & 7 = 1, display 384, 1
nop
nop
nop
nop
nop
nop

# PC = 408
ori $t7, $t7, 2        # t7 = 7 | 2 = 7, display 408, 7
nop
nop
nop
nop
nop
nop

# PC = 432
xori $t9, $t9, 1        # t9 = 6 ^ 1 = 7, display 432, 7
nop
```

```
nop
nop
nop
nop

#####
# Shift Instructions
#####

# PC = 456
sll $s3, $t1, 2      # s3 = 3 << 2 = 12, display 456, 12
nop
nop
nop
nop
nop
nop

# PC = 480
srl $s4, $s3, 1      # s4 = 12 >> 1 = 6, display 480, 6
nop
nop
nop
nop
nop
nop

#####
# Set Instructions
#####

# PC = 504
slt $s5, $t1, $t0      # s5 = (3 < 5) = 1, display 504, 1
nop
nop
nop
nop
nop
nop

# PC = 528
slti $s6, $t0, 10      # s6 = (5 < 10) = 1, display 528, 1
nop
nop
nop
nop
nop
nop
```

```
#####
# Branch & Jump Instructions
#####
```

```
# PC = 552
addi $t0, $zero, 2      # t0 = 2, display 552, 2
nop
nop
nop
nop
nop
nop
```

```
# PC = 576
bgez $t0, label_bgez    # branch (true, t0 >= 0), display 576, 0
nop
nop
nop
nop
nop
nop
```

```
(600)
addi $t0, $zero, 0      # skipped
nop
nop
nop
nop
nop
nop
```

```
Label_bgez:
(624)
# PC = 600
beq $t0, $t0, label_beq  # branch (true), display 600, 0
nop
nop
nop
nop
nop
nop
```

```
(648)
addi $t1, $zero, 9      # skipped
nop
nop
nop
```

```
nop
nop

Label_beq:
(672)
# PC = 624
    bne $t0, $t1, label_bne  # branch (true, 2 != 3), display 624, 0
    nop
    nop
    nop
    nop
    nop
    nop

(696)
    addi $t2, $zero, 10      # skipped
    nop
    nop
    nop
    nop
    nop
    nop

Label_bne:
(720)
# PC = 648
    bgtz $t0, label_bgtz   # branch (true, 2 > 0), display 648, 0
    nop
    nop
    nop
    nop
    nop
    nop

(744)
    addi $t3, $zero, 11      # skipped
    nop
    nop
    nop
    nop
    nop
    nop

(768)
label_bgtz:
# PC = 672
    blez $t1, label_blez    # (3 <= 0) false, display 672, 0
    nop
```

```
nop
nop
nop
nop

(792)
# PC = 696
    addi $t4, $zero, 12      # executes, display 696, 12
    nop
    nop
    nop
    nop
    nop
    nop

(816)
label_blez:
# PC = 720
    bltz $t1, label_bltz    # (3 < 0) false, display 720, 0
    nop
    nop
    nop
    nop
    nop
    nop

(840)
    addi $t5, $zero, 13      # executes, display 744, 13
    nop
    nop
    nop
    nop
    nop
    nop

(864)
label_bltz:
    j label_jump            # jump unconditionally, display 768, 0
    nop
    nop
    nop
    nop
    nop
    nop

(888)
label_jump:
    jal label_link          # jump and link, display 792, 0
```

```
nop  
nop  
nop  
nop  
nop  
  
(912)  
    addi $t6, $zero, 14      # skipped  
    nop  
    nop  
    nop  
    nop  
    nop  
    nop
```

```
(936)  
label_link:  
    jr $ra                  # return from jal, display 816, 0  
    nop  
    nop  
    nop  
    nop  
    nop  
    nop
```

```
#####
# Infinite Loop for Testing
#####
```

```
(960)  
# PC = 840  
    j main                 # restart test sequence, display 840, 0  
    nop  
    nop  
    nop  
    nop  
    nop  
    nop
```