Benchmark: “Max & Min”

Author: Hassan TaqiEddin *Reviewed by: “Omar Alsweiti”*

Description & Notes

- Find the maximum and minimum value of array.
- Contains one loop and tested variety of instructions.
- Could be used to test many iterations (O(n) complexity)

Algorithm (Pseudo or C)

```
for i <- 1 to size - 1
  do  key <- Array[i]
   if (key > max)
     max = key
     continue
   if (key < min)
     min = key
```

Registers and memory used in implementation

$2 : i (loop index)

$5 : Temporary register for calculating array offsets (address for Array[i])

$10 : max (holds the maximum value)

$15 : min (holds the minimum value)

$16 : Temporary register for the value of Array[i]

$20 : size (size of the Array)

$25 : temp for condition

$26 : temp for condition

$27 : temp for condition

Code (.data and .text)

```
  .data
Array: .word 0x10, 0xF, 0x5, 0x9, 0x20, 0x19, 0x4, 0x1E, 0x9, 0xB

    .text
main:
    # Initialize registers
    ORI  $2 , $0, 0x0       # $2 = 0
    ADDI $20, $0, 0xA       # $20 = array size (10)
    XORI $31, $0, 0x1       # $31 = 1
    ANDI $5 , $0, 0x0       # $5 = 0
    LW   $10, 0x0($5)       # $10 = Array[0]
    LW   $15, 0x0($5)       # $15 = Array[0]

LOOP:
    ADDI $2,  $2, 1         # Increment i
    SGT  $25, $20, $2       # Check if size > i
    BNE  $25, $31, END      # Exit loop if i >= size

    # Choose one of these Insertion based on your memory
    # For Word addressable        # For byte addressable
    # ADD  $5, $2, $0             # SLL  $5, $2, 2

    LW   $16, 0x0($5)       # $16 = Arr[i]
    SGT  $26, $16, $10      # $26 = 1 if Arr[i] > max
    BEQ  $26, $0, MIN       # Skip updating max if condition is false
    OR   $10, $16, $0       # Update max
    J LOOP

MIN:
    # Check if current element is less than min
    SLT  $27, $16, $15      # $27 = 1 if Arr[i] < min
    BEQ  $27, $0, LOOP      # Skip updating min if condition is false
    ADD  $15, $16, $0       # Update min
    J    LOOP               # Repeat loop

END:
     NOP # (NOP equals to SLL $0, $0, 0)
```

Expected Output

$10 = 0x20 # Max Number

$15 = 0x04 # Min Number