Benchmark: “Swapping”

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

Description & Notes

- This Benchmark is used to swap two registers values.
- Immediate values are in decimal.
- All memory addresses should be set to **zero**.
- Registers values are set 0.
- After each swap, choose the right store instruction based on your memory (Byte or Word) addressable.
- The final values locations are different between word and byte addressable memories.
- Three types of swapping are implemented:
    - Swapping with temp.
    - Swapping using adding without temp.
    - Swapping using Xoring without temp.

Registers and memory used in implementation

$2, $3, $4, $5, $6, $7

Are registers used to be swapped between.

$31: Temp for swapping

Code (.data and .text)

```
    .text
main:
    # Initialize registers
    ADDI $2 , $0, 5        # $2 = 5
    ADDI $3 , $0, 10       # $3 = 10
    ANDI $31, $0, 0x0   # $31 = 0
    ADD  $31 ,$0, $3
    ADD  $3 , $0, $2
    ADD  $2 , $0, $31

    # For word addressable      # For Byte Addressable
    # Sw  $2, 0x1($0)           Sw  $2, 0x4($0)
    # Sw  $3, 0x2($0)           Sw  $3, 0x8($0)

    ADDI $4 , $0, 15    # $4 = 15
    ADDI $5 , $0, 20    # $5 = 20
    ADD  $5 , $4, $5
    SUB  $4 , $5, $4
    SUB  $5 , $5, $4

    # For word addressable      # For Byte Addressable
    # Sw  $4, 0x3($0)           Sw  $4, 0xb($0)
    # Sw  $5, 0x4($0)           Sw  $5, 0x10($0)

    ADDI $6, $0 , 25    # $6 = 25
    ADDI $7, $0 , 30    # $7 = 30
    XOR  $6, $6 , $7
    XOR  $7, $6 , $7
    XOR  $6, $6 , $7

    # For word addressable      # For Byte Addressable
    # Sw  $6, 0x5($0)           Sw  $6, 0x14($0)
    # Sw  $7, 0x6($0)           Sw  $7, 0x18($0)

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

Expected Output

$2: 10
$3: 5
$4: 20
$5: 15
$6: 30
$7: 25
$31: 10

# For word addressable you endian)	# For byte addressable (Check
Mem [1]: 0x000a	Mem [4:7]: 0x000a
Mem [2]: 0x0005	Mem [8:b]: 0x0005
Mem [3]: 0x0014	Mem [c:f]: 0x0014
Mem [4]: 0x000f	Mem [10:13]: 0x000f
Mem [5]: 0x001e	Mem [14:17]: 0x001e
Mem [6]: 0x0019	Mem [18:1b]: 0x0019