Benchmark: “Benchmark Name”

Author: Abdalrahman Ebdah *Reviewed by: “Omar Alsweiti”*

Description & Notes

- Binary Search Algorithm
- Contain a while loop and use multiple comparisons to break the loop
- Perform integer division using SLR instruction to calculate the mid pointer
- Benchmark Complexity: O(logN)
- Support both word and byte addressing

Algorithm (Pseudo or C)

```
while (right <= left) {
      mid = (right + left)/2;
      if(a[mid]==searchElement){
            answer = mid;
            break;
      }
      else if(a[mid] > searchElement)
            left = mid+1;

      else  right = mid-1;
}
```

Code (.data and .text)

```
    .data
myArray:    .word 1, 4, 5, 7, 9, 12, 15, 17, 20, 21, 30 arraySize: .word 11

    .text #Initailiztion addi $1, $0, 0x0
addi $2, $0, 0xB
addi $3, $0, 0x7
loop:     slt $7, $2, $1
          bne $0, $7, notFound
          add $4, $2, $1
          srl $5, $4, 1
          #sll $5, $5, 2 For byte addressable memory
          lw  $6, 0x0($5)
          #srl $5, $5, 2 For byte addressable memory
          beq $3, $6, found
          slt $6, $6, $3
          beq $6, $0, leftHalf
          j   rightHalf
leftHalf:
          add $2, $5, 0xFFFF #"FFFF=1"
          j loop
rightHalf:
          addi $1, $5, 0x1
          j loop
found:
          add $8, $0, $5
          j finish
notFound:
          addi $8, $0, 0xFFFF
          j finish
finish:
          NOP
```

Expected Output

$8 = searchElement