

ArraysAndMIPS

April 24, 2023

1 Compile the code into MIPS assembly

```
int i;
int array[1000];
for(i = 0; i < 100; i++)
    array[i] = array[i] * 8;
```

Assumptions:

- Assume that `i` corresponds to: `$s0` and the 100 value in the loop is stored in `$s2`
- Assume that array base address corresponds to: `$s1`

The execution will be as follows: `array[i]` gets loaded from the memory, then is multiplied by 8 and then it will be stored back to memory, so we need `array[i]` address, before we can add `i` to the base address of the `array` we have to multiply its value by 4 due to byte addressing of MIPS.

Then we will multiply the value by 8 (with shifting left 3 times), and we store back the value into the same address.

Assembly starts here:

```
and $s0, $s0, $zero    # int i = 0, and will make it zero 'cause it is anded with $zero
```

Loop:

```
sll $t0, $s0, 2        # Temp reg $t0 = i * 4
add $t0, $t0, $s1       # $t0 = address of array[i]
lw $t1, 0($t0)          # Temp reg $t1 = array[i]
sll $t1, $t1, 3         # $t1 = $t1 * 8 (or * 2^3 so shift three times to left)
sw $t1, 0($t0)          # array[i] <- array[i]
addi $s0, $s0, 1        # i++
beq $s0, $s2, Exit      # Loop will be terminated
j Loop                  # Loop is not terminated and we will start again
```

Exit:

...