Module: R4: Computer Architecture
Section: Caches **Task:** Memory Accesses

Task 2

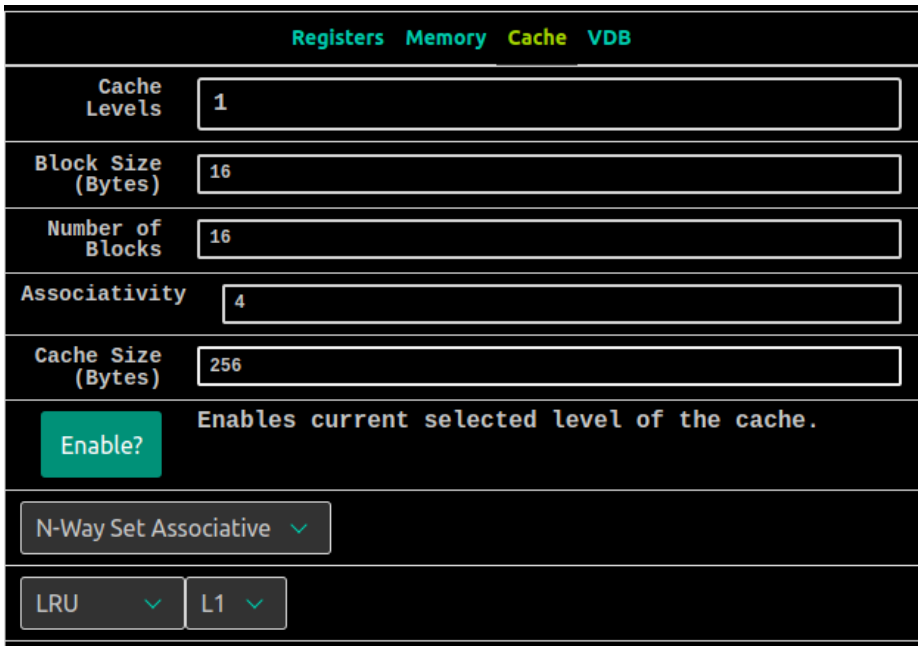
Memory Accesses

Scenario 2

1. **Program Parameters:**

```
# You MAY change the code below this section
main:   li  a0, 256     # array size in BYTES (power of 2 < array size)
    li  a1, 2       # step size  (power of 2 > 0)
    li  a2, 1       # rep count  (int > 0)
    li  a3, 1       # 0 - option 0, 1 - option 1
# You MAY change the code above this section
```

2. **Cache Parameters:**



➢ **Questions:**

1. ***How many memory accesses are there per iteration of the inner loop (not the one involving Rep Count)?***

   Hit Count: 48
   Accesses: 64

   Since each iteration of the inner loop involves two memory accesses (read and write), we can calculate the number of memory accesses per iteration:

Memory accesses per iteration = $\frac{Total\ Accesses}{2 * rep}$

Memory accesses per iteration = $\frac{64}{2 * 1}$ = 32

2. *What is the repeating hit/miss pattern?*

The repeating hit/miss pattern can be determined from the visualization. We observe from the accesses and the hit count which demonstrates that every 4 accesses result in 3 hits and 1 miss. Henceforth, the hit rate of 0.75.

The repeating pattern is ***mhhh***.

3. *Keeping everything else the same, what does our hit rate approach as Rep Count goes to infinity?*

As we increase our rep-count, our Hit Rate approaches to 1 which can be calculated from the following formula:

Assuming: Rep-Count = 88

Hit Rate = $\frac{Hit\ Count}{Total\ Accesses}$ = $\frac{5612}{64 * 88}$ = 0.99644 ≅ 1.0

4. *Fill in the BLANKS*

We should try to access one [**segment**] of the array at a time and apply all of the [**functions**] to that [**segment**] so we can be completely done with it before moving on, thereby keeping that [**segment**] hot in the cache and not having to circle back to it later on.