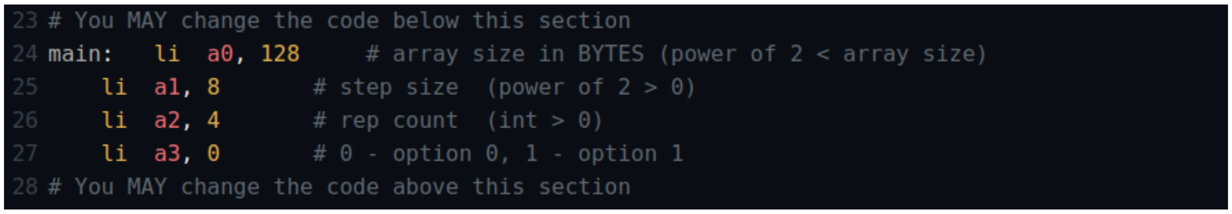
Module: R4: Computer Architecture
Section: Caches **Task:** Memory Accesses

Task 2

Memory Accesses

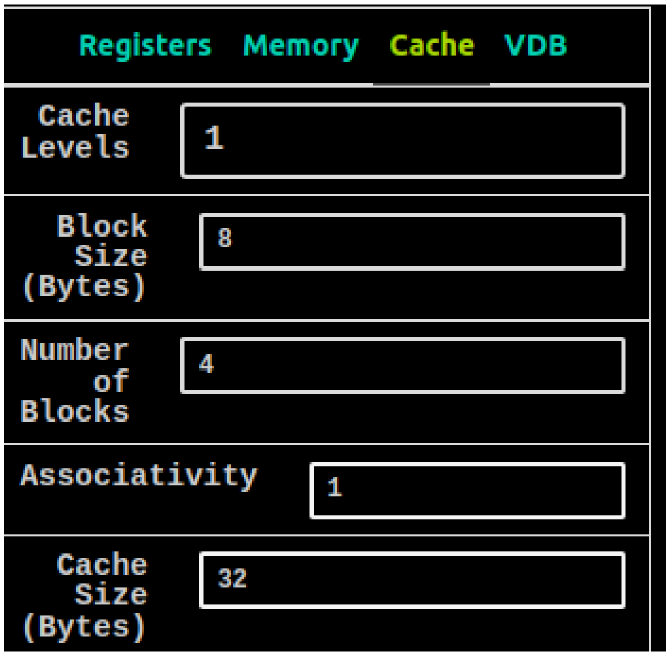
Scenario 1

1. **Program Parameters:**



```
# You MAY change the code below this section
main:   li  a0, 128     # array size in BYTES (power of 2 < array size)
    li  a1, 8       # step size  (power of 2 > 0)
    li  a2, 4       # rep count  (int > 0)
    li  a3, 0       # 0 - option 0, 1 - option 1
# You MAY change the code above this section
```

2. **Cache Parameters:**



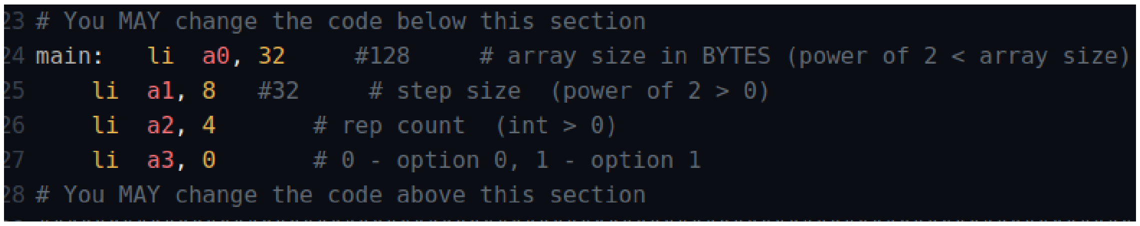
Cache Levels	1
Block Size (Bytes)	8
Number of Blocks	4
Associativity	1
Cache Size (Bytes)	32

➢ **Questions:**

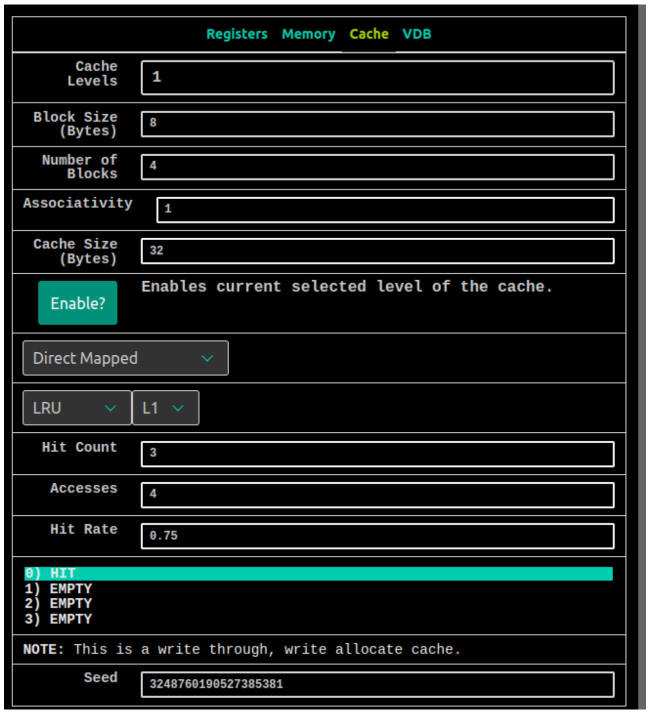
1. ***What combination of parameters is producing the hit rate you observe?***
   The combination of parameters that is producing the hit rate are **"step-size and block size"**. Because step-size in bytes is exactly equal to the block size in bytes (8 bytes). So, whenever we try to write to a block while it is warmed up, it results into a conflict miss because we have updated the index to (index = index + step-size). So it always tries to write a different value to the same index(0th row) in cache which results into a conflict miss. So we are missing everytime. **Hit Rate = 0.0**

2. ***What is the hit rate if we increase Rep Count arbitrarily?***
   Increasing rep-count arbitrarily won't affect the hit rate because there were no hits in the cache in the first place. As rep-count increases, the program will still fail to find data in the cache and will result in misses. Therefore, the hit rate will remain 0.

3. ***How could we modify one program parameter to get an increased hit rate?***
   We can actually do it by 2 ways.
   I. `[a0] = 32 bytes` ***#by changing array size***
   II. `[a1] = 32 bytes` ***#by changing step-size***



```
# You MAY change the code below this section
main:   li  a0, 32     #128     # array size in BYTES (power of 2 < array size)
    li  a1, 8   #32     # step size  (power of 2 > 0)
    li  a2, 4       # rep count  (int > 0)
    li  a3, 0       # 0 - option 0, 1 - option 1
# You MAY change the code above this section
```



Cache Levels: 1
Block Size (Bytes): 8
Number of Blocks: 4
Associativity: 1
Cache Size (Bytes): 32
Enables current selected level of the cache.
Direct Mapped
LRU L1
Hit Count: 3
Accesses: 4
Hit Rate: 0.75

```
0) HIT
1) EMPTY
2) EMPTY
3) EMPTY
```

NOTE: This is a write through, write allocate cache.
Seed: 3248760190527385381