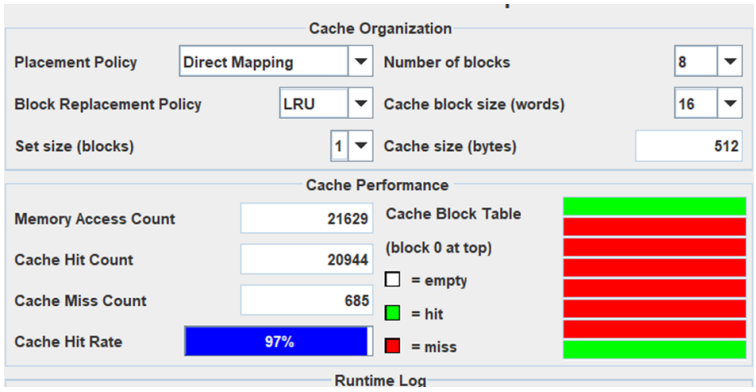
1. **Find the hit rates for each cache configurations.**

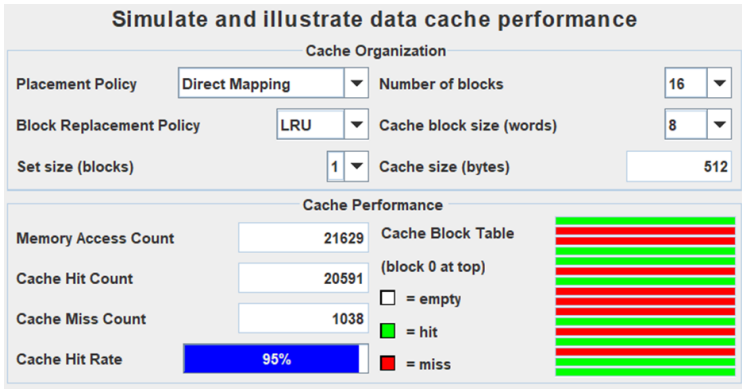
Note: Here, for some reason, we wouldn't be able to to simulate the first configuration because mips doesn't support analysis of such cache size as we researched.

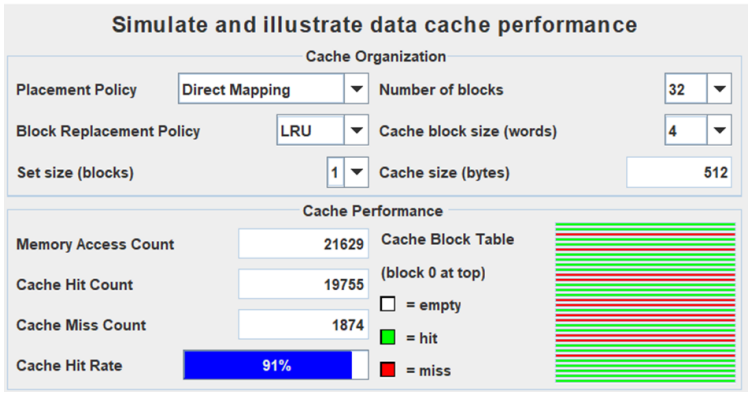
For each configuration with direct mapping:

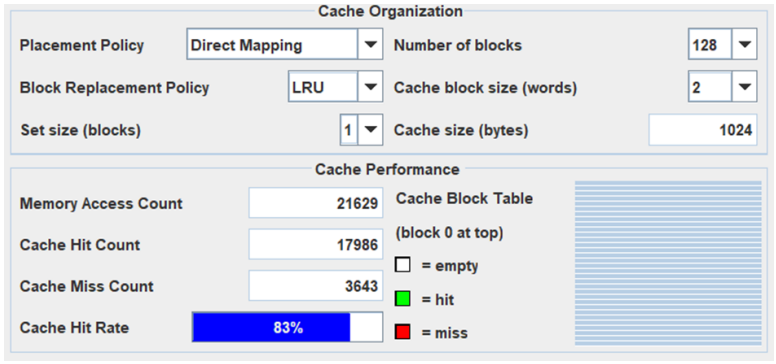
Block Size	No. of Blocks	Cache Size	Hit Rate
4 B	512	2048 B	<%83
8 B	128	1024 B	%83
16 B	32	512 B	%91
32 B	16	512 B	%95
64 B	8	512 B	%97

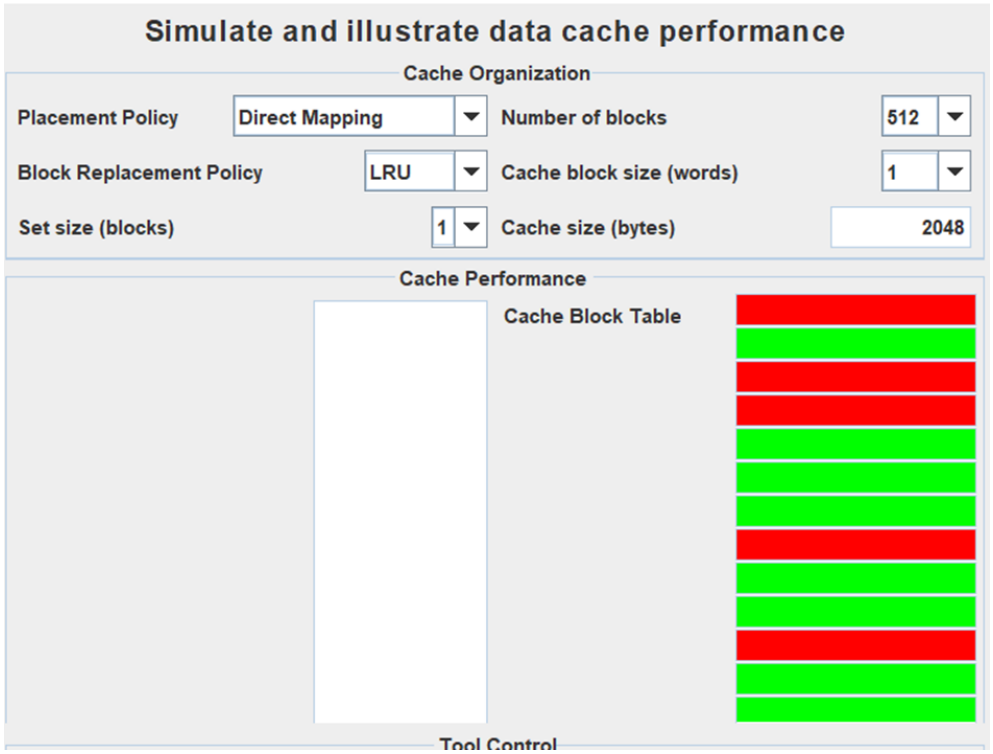
Full Results:






Simulate and illustrate data cache performance
Cache Organization
Placement Policy
Direct Mapping
Number of blocks
32
Block Replacement Policy
LRU
Cache block size (words)
4
Set size (blocks)
1
Cache size (bytes)
512
Cache Performance
Memory Access Count
21629
Cache Hit Count
19755
Cache Miss Count
1874
Cache Hit Rate
91%
Cache Block Table
(block 0 at top)
= empty
= hit
= miss


Cache Organization
Placement Policy
Direct Mapping
Number of blocks
128
Block Replacement Policy
LRU
Cache block size (words)
2
Set size (blocks)
1
Cache size (bytes)
1024
Cache Performance
Memory Access Count
21629
Cache Hit Count
17986
Cache Miss Count
3643
Cache Hit Rate
83%
Cache Block Table
(block 0 at top)
= empty
= hit
= miss


Simulate and illustrate data cache performance
Cache Organization
Placement Policy
Direct Mapping
Number of blocks
512
Block Replacement Policy
LRU
Cache block size (words)
1
Set size (blocks)
1
Cache size (bytes)
2048
Cache Performance
Cache Block Table
Tool Control

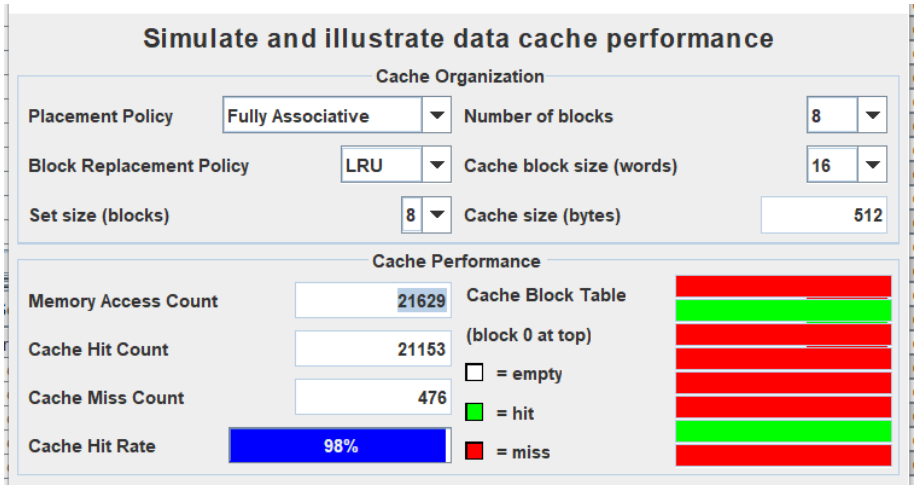
Fully Associative:

For each configuration with fully Associative:

Block Size	No. of Blocks	Cache Size	Hit Rate
4 B	512	2048 B	<%83
8 B	128	1024 B	%83
16 B	32	512 B	%92
32 B	16	512 B	%96
64 B	8	512 B	%98

2. Which cache configuration yields the best performance?

Max Hit rate is fully associative with 8 blocks and 16 words(64 bytes) of block size as can be seen below with a hit rate of 98%.



3. Does the largest cache always give the best result? Why or why not?

No. This is because of three crucial factors: latency, cache hit rate versus size, and replacement policies. Firstly, larger caches inherently have longer latency due to the physical distance the data must travel, which can slow down the CPU's access speed. This increased latency could offset the advantages provided by the increased storage capacity of a larger cache. Secondly, although an increase in cache size can initially improve the cache hit rate, there is a point of diminishing returns. Once the cache reaches a certain size, the hit rate no longer improves significantly. This happens because larger caches start holding data that is less likely to be accessed in the near future. Finally, the replacement policy, which determines what data to evict when the cache is full, greatly influences a cache's effectiveness. A larger cache with a poor

replacement policy might experience more cache misses than a smaller cache with a more effective policy. So, while a larger cache has its benefits, these must be balanced with considerations of latency, hit rates, and replacement policies for optimal performance.