Computer Organization and Assembly Language Programming

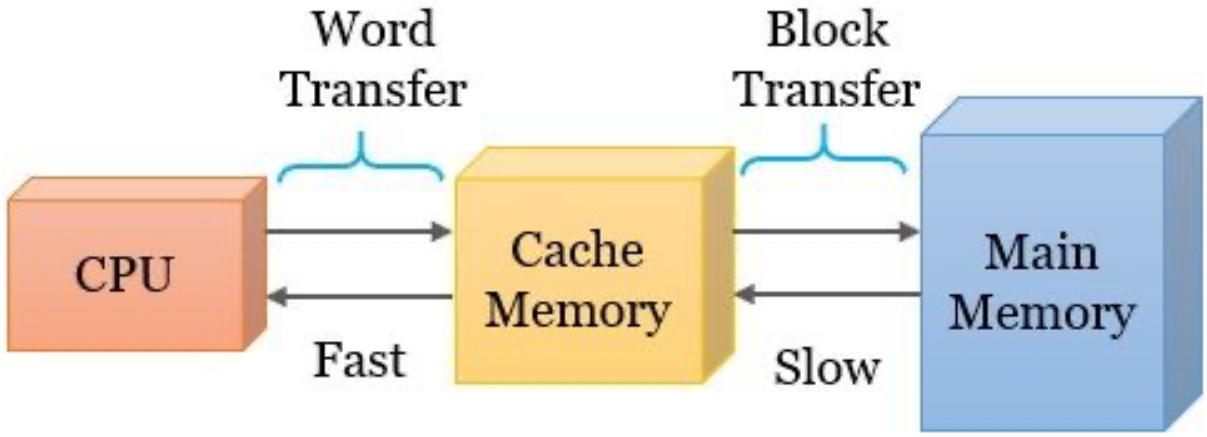
CSCE-231

Summer 2024

Project 2: Cache Simulator - Report

Jana Fadl

Introduction:



Cache memory is a small, high-speed storage area located between the CPU and the main memory, designed to store frequently accessed data and instructions to speed up processing. Direct-mapped cache is a simple and efficient type of cache where each block of main memory maps to exactly one cache line, making it easy to implement but prone to conflicts if multiple blocks compete for the same cache line. Fully associative cache, on the other hand, allows any block of main memory to be stored in any cache line, providing flexibility and reducing conflicts but at the cost of more complex hardware and slower access times. To simulate these 2 types of caches I will be using C++ to create models that track memory accesses and the state of cache lines. Direct-mapped cache can be simulated using an array where each index represents a cache line. fully associative cache can be modeled using an array as well with additional logic to manage the placement and replacement of cache lines based on access patterns. Fully associative cache can be implemented in several ways which include using a random replacement policy by utilizing a random number generator to decide which cache line to replace when a new block of data needs to be loaded into the cache.

1. Initialize the Cache: Represent the cache using a list or array structure where each element corresponds to a cache line. Each cache line can store data, a tag, and a valid bit to indicate if the cache line contains valid data.

2. Random Number Generator: Use a random number generator function to produce indices that correspond to cache lines. This function will be used to select a random cache line for replacement when needed.

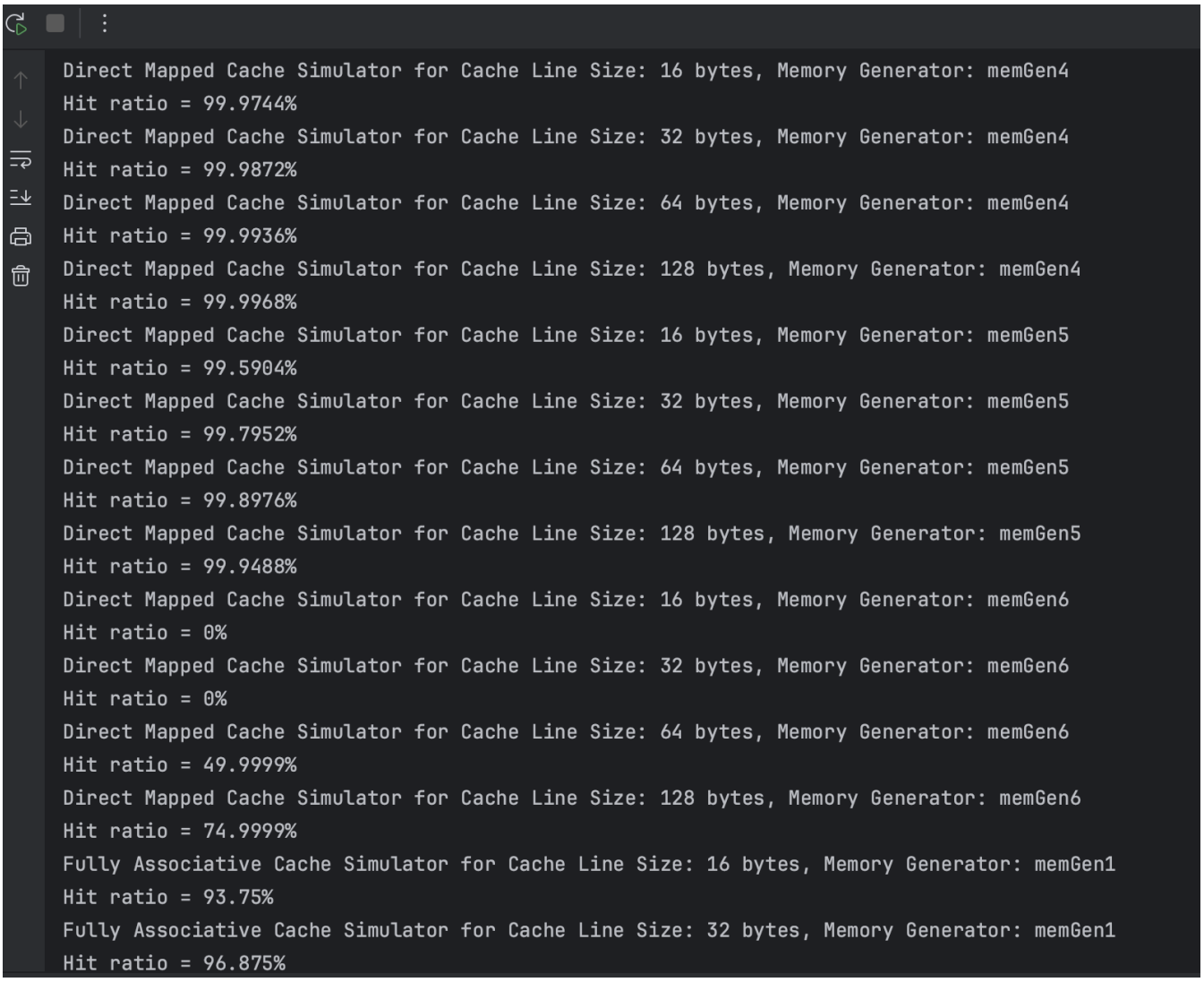
3. Accessing Cache:
   - Check for Hit: When accessing data, compute the tag for the desired block and search through the cache lines to find a match.
   - Handle Miss: If the data is not found in the cache (a cache miss), proceed to the replacement step.

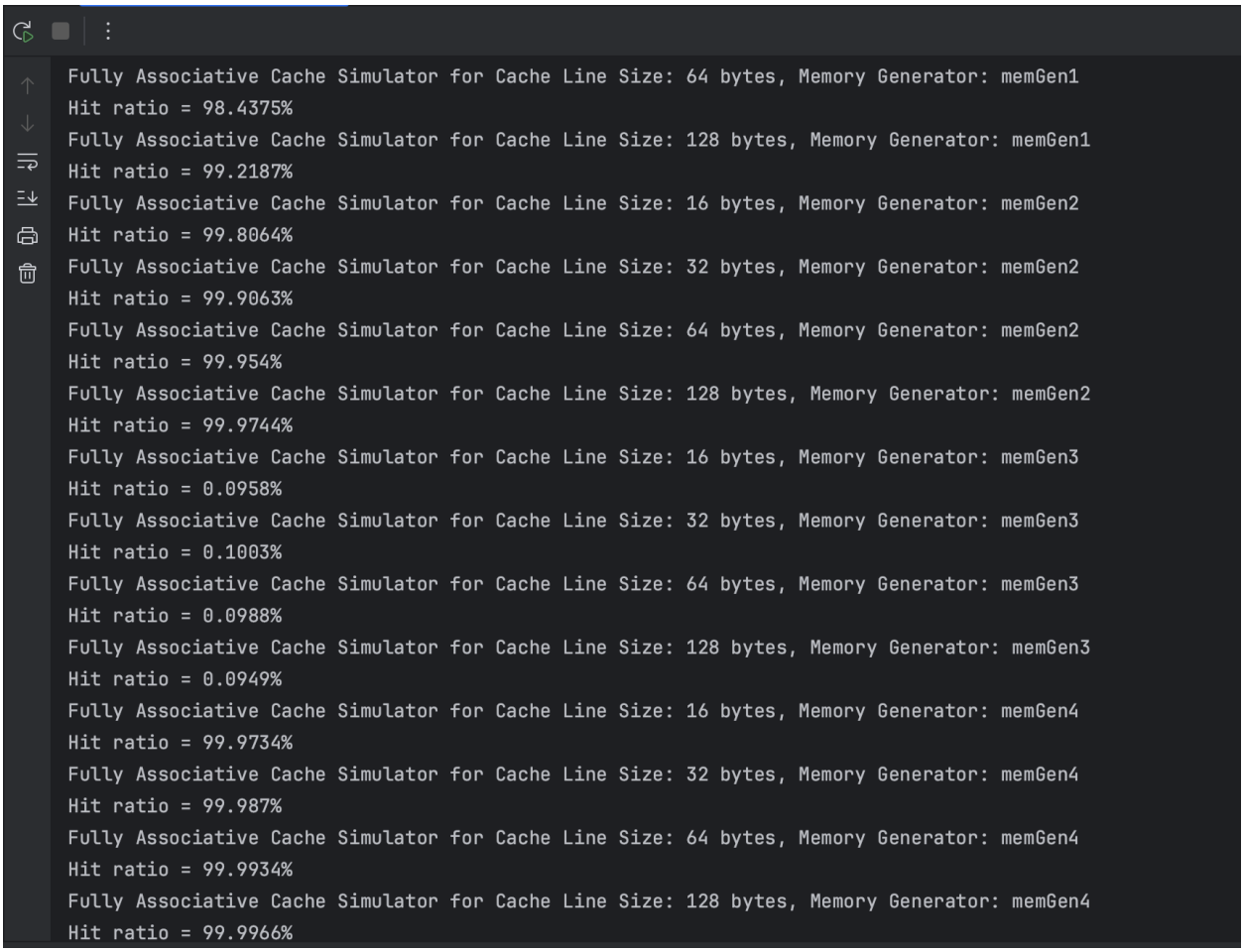
4. Replacement:

- Random Replacement: Generate a random index within the range of cache line indices. The cache line at this randomly selected index will be replaced with the new block of data.
- Update Cache Line: Update the selected cache line with the new block's data, tag, and set the valid bit to true.

Experiment Results:

```
/Users/janafadl/Documents/GitHub/CacheSimulator/CacheSimulator
Direct Mapped Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen1
Hit ratio = 93.75%
Direct Mapped Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen1
Hit ratio = 96.875%
Direct Mapped Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen1
Hit ratio = 98.4375%
Direct Mapped Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen1
Hit ratio = 99.2187%
Direct Mapped Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen2
Hit ratio = 99.8464%
Direct Mapped Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen2
Hit ratio = 99.9232%
Direct Mapped Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen2
Hit ratio = 99.9616%
Direct Mapped Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen2
Hit ratio = 99.9808%
Direct Mapped Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen3
Hit ratio = 0.0959%
Direct Mapped Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen3
Hit ratio = 0.0946%
Direct Mapped Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen3
Hit ratio = 0.0951%
Direct Mapped Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen3
Hit ratio = 0.0971%
Direct Mapped Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen4
Hit ratio = 99.9744%
Direct Mapped Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen4
```

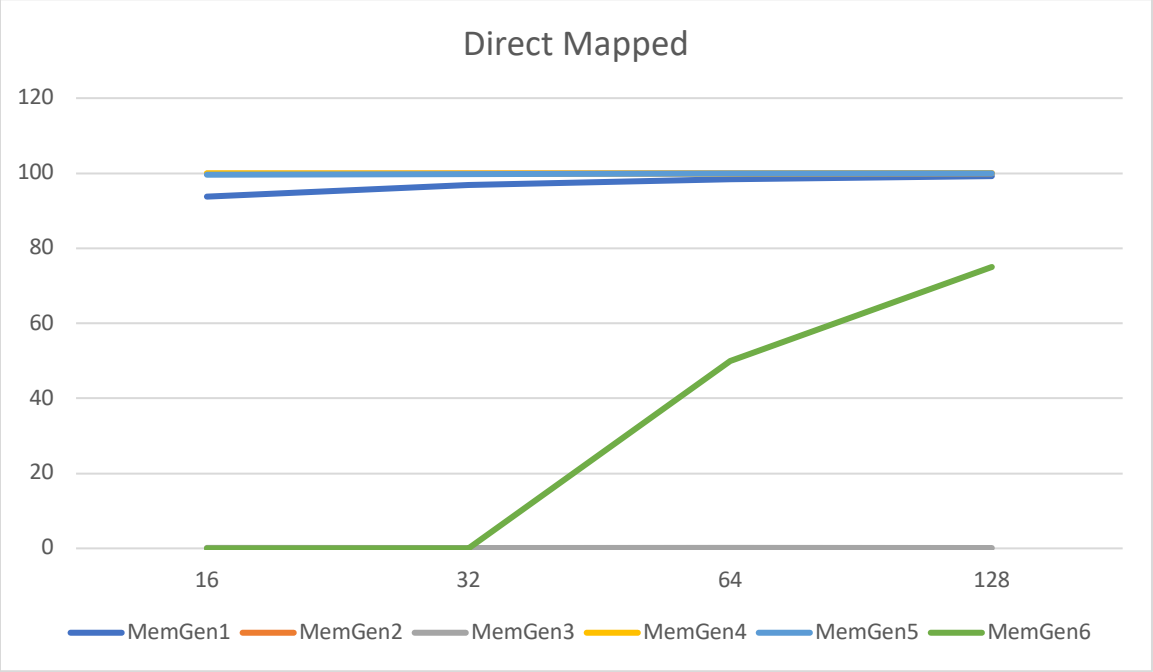

Direct Mapped Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen4
Hit ratio = 99.9744%
Direct Mapped Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen4
Hit ratio = 99.9872%
Direct Mapped Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen4
Hit ratio = 99.9936%
Direct Mapped Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen4
Hit ratio = 99.9968%
Direct Mapped Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen5
Hit ratio = 99.5904%
Direct Mapped Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen5
Hit ratio = 99.7952%
Direct Mapped Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen5
Hit ratio = 99.8976%
Direct Mapped Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen5
Hit ratio = 99.9488%
Direct Mapped Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen6
Hit ratio = 0%
Direct Mapped Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen6
Hit ratio = 0%
Direct Mapped Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen6
Hit ratio = 49.9999%
Direct Mapped Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen6
Hit ratio = 74.9999%
Fully Associative Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen1
Hit ratio = 93.75%
Fully Associative Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen1
Hit ratio = 96.875%

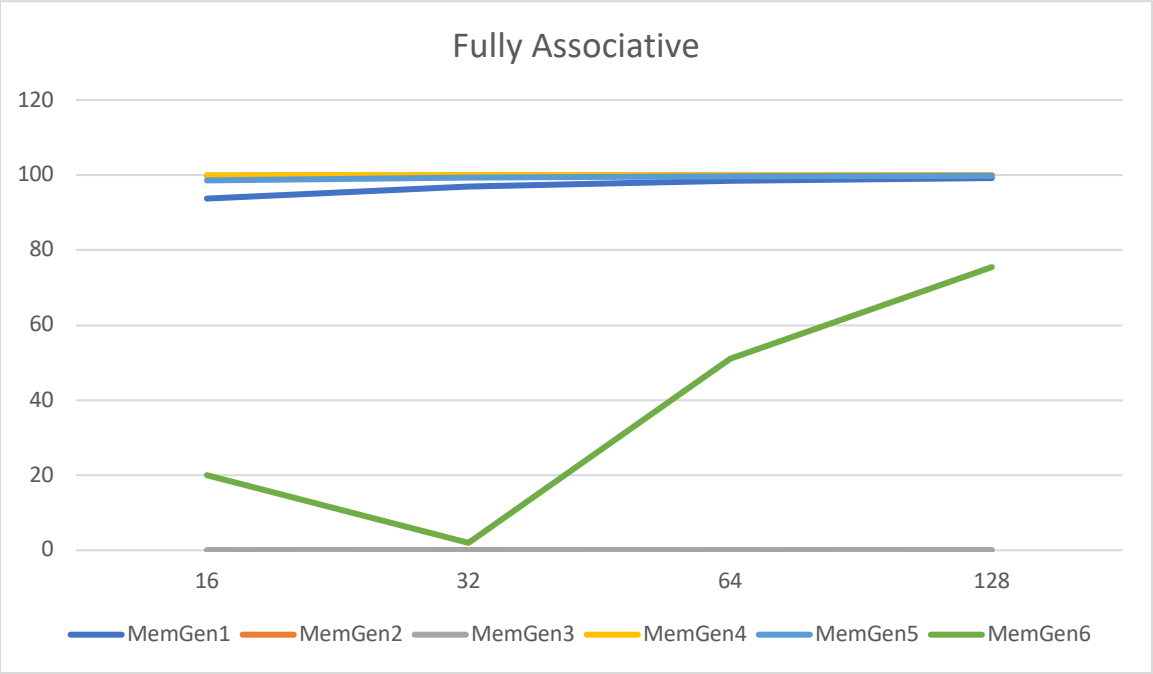

Fully Associative Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen1
Hit ratio = 98.4375%
Fully Associative Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen1
Hit ratio = 99.2187%
Fully Associative Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen2
Hit ratio = 99.8064%
Fully Associative Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen2
Hit ratio = 99.9063%
Fully Associative Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen2
Hit ratio = 99.954%
Fully Associative Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen2
Hit ratio = 99.9744%
Fully Associative Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen3
Hit ratio = 0.0958%
Fully Associative Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen3
Hit ratio = 0.1003%
Fully Associative Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen3
Hit ratio = 0.0988%
Fully Associative Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen3
Hit ratio = 0.0949%
Fully Associative Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen4
Hit ratio = 99.9734%
Fully Associative Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen4
Hit ratio = 99.987%
Fully Associative Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen4
Hit ratio = 99.9934%
Fully Associative Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen4
Hit ratio = 99.9966%

```
Hit ratio = 0.0949%
Fully Associative Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen4
Hit ratio = 99.9734%
Fully Associative Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen4
Hit ratio = 99.987%
Fully Associative Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen4
Hit ratio = 99.9934%
Fully Associative Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen4
Hit ratio = 99.9966%
Fully Associative Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen5
Hit ratio = 98.5869%
Fully Associative Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen5
Hit ratio = 99.2885%
Fully Associative Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen5
Hit ratio = 99.6469%
Fully Associative Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen5
Hit ratio = 99.8295%
Fully Associative Cache Simulator for Cache Line Size: 16 bytes, Memory Generator: memGen6
Hit ratio = 20.0973%
Fully Associative Cache Simulator for Cache Line Size: 32 bytes, Memory Generator: memGen6
Hit ratio = 1.988%
Fully Associative Cache Simulator for Cache Line Size: 64 bytes, Memory Generator: memGen6
Hit ratio = 50.9874%
Fully Associative Cache Simulator for Cache Line Size: 128 bytes, Memory Generator: memGen6
Hit ratio = 75.4877%

Process finished with exit code 0
```

Graphs Of Results:





Analysis:
The code generates memory addresses using six different memory address generators, each implementing a different strategy for address generation.

MemGen1

- Generates sequential addresses.
- Uses a static variable addr which is incremented with each call and returns addr % DRAM_SIZE to wrap around after reaching the size of DRAM.

- The expected hit rate for this function is **high** because sequential access patterns have high spatial locality. Once the cache is filled initially, subsequent accesses are likely to hit unless the sequence spans more addresses than the cache can hold, causing it to wrap around. The direct-mapped cache might have lower hit rates if multiple sequential blocks map to the same cache line.

MemGen2

- Generates random addresses within a 24 KB range.
- Calls the rand_() function to get a random number and returns it modulo 24 * 1024 to keep the addresses within 24 KB.

- The expected hit rate for this function is **moderate to high** since the address range is limited to 24 KB and the cache size is 64 KB, the entire range can fit into the cache. However, due to randomness, some cache lines may be underutilized while others may be frequently replaced,

reducing the hit rate. Fully associative cache should have a higher hit rate than direct-mapped cache due to its flexibility in placement.

MemGen3

- Generates completely random addresses within the full DRAM size.
- Calls the rand_() function to get a random number and returns it modulo DRAM_SIZE.

- The expected hit rate for this function is **low** due to random addresses across a large space leading to poor spatial and temporal locality. Each access is likely to map to a different cache line, leading to many misses. Both direct-mapped and fully associative caches will experience frequent cache line replacements.

MemGen4

- Generates sequential addresses within a 4 KB range.
- Uses a static variable addr which is incremented with each call and returns addr % (4 * 1024) to wrap around after reaching 4 KB.

- The expected hit ratio for this function is **very high/ high**. Similar to memGen1 but with a smaller range, ensuring that the entire range fits well within the cache (64 KB). This generator takes advantage of spatial locality, and after the initial misses, subsequent accesses should hit consistently.

MemGen5

- Generates sequential addresses within a 64 KB range.
- Uses a static variable addr which is incremented with each call and returns addr % (1024 * 64) to wrap around after reaching 64 KB.

- The expected hit ratio here would most likely be **high** as well as the address range matches the cache size exactly. Sequential access ensures high spatial locality. Once the entire range is loaded into the cache, subsequent accesses will mostly hit. Direct-mapped cache might see conflicts if multiple blocks map to the same line.

MemGen6

- Generates addresses with a fixed stride of 32 bytes within a 256 KB range.
- Uses a static variable addr which is incremented by 32 with each call and returns addr % (64 * 4 * 1024) to wrap around after reaching 256 KB.

- For this function it is expected that the hit ratio would be **moderate** as only every 32nd byte is accessed. This pattern might result in lower utilization of cache lines. As the range is 256 KB and the cache is 64 KB, not all accessed blocks will fit in the cache, leading to frequent replacements. Fully associative cache should handle this pattern better than direct-mapped cache.

Conclusion:

Based on the results of the experiment, the graphs and analysis of functions we can conclude that the results mostly fit the expectations of each function.

The results show higher hit rates for MemGen1, MemGen2, MemGen4, MemGen5

Moderate hit rates for MemGen6

And lower hit rates for MemGen3

With some differences between the direct mapped cache and the fully associative cache due to the different handling methods for each.

Test Case Result:

```
testAddresses[] = { 0x0000, 0x0004, 0x0010, 0x00FF, 0x0100, 0x1000, 0xFFFF,
0xABCD, 0x1234, 0xFFFFF }
```

```
Direct Mapped Cache Simulator for Cache Line Size: 16 bytes with predefined addresses
0x00000000 (Miss)
0x00000004 (Hit)
0x00000010 (Miss)
0x000000ff (Miss)
0x00000100 (Miss)
0x00001000 (Miss)
0x0000ffff (Miss)
0x0000abcd (Miss)
0x00001234 (Miss)
0x000fffff (Miss)
Hit ratio = 10%
```

```
Direct Mapped Cache Simulator for Cache Line Size: 20 bytes with predefined addresses
0x00000000 (Miss)
0x00000004 (Hit)
0x00000010 (Hit)
0x000000ff (Miss)
0x00000100 (Miss)
0x00001000 (Miss)
0x0000ffff (Miss)
0x0000abcd (Miss)
0x00001234 (Miss)
0x000fffff (Miss)
Hit ratio = 20%
```

```
Direct Mapped Cache Simulator for Cache Line Size: 40 bytes with predefined addresses
0x00000000 (Miss)
0x00000004 (Hit)
0x00000010 (Hit)
0x000000ff (Miss)
0x00000100 (Miss)
0x00001000 (Miss)
0x0000ffff (Miss)
0x0000abcd (Miss)
0x00001234 (Miss)
0x000fffff (Miss)
Hit ratio = 20%
```

```
Direct Mapped Cache Simulator for Cache Line Size: 80 bytes with predefined addresses
0x00000000 (Miss)
0x00000004 (Hit)
0x00000010 (Hit)
0x000000ff (Miss)
0x00000100 (Miss)
0x00001000 (Miss)
0x0000ffff (Miss)
0x0000abcd (Miss)
0x00001234 (Miss)
0x000fffff (Miss)
Hit ratio = 20%
```

```
Fully Associative Cache Simulator for Cache Line Size: 10 bytes with predefined addresses
0x00000000 (Miss)
0x00000004 (Hit)
0x00000010 (Miss)
0x000000ff (Miss)
0x00000100 (Miss)
0x00001000 (Miss)
0x0000ffff (Miss)
0x0000abcd (Miss)
0x00001234 (Miss)
0x000fffff (Miss)
Hit ratio = 10%
```

```
Fully Associative Cache Simulator for Cache Line Size: 20 bytes with predefined addresses
0x00000000 (Miss)
0x00000004 (Hit)
0x00000010 (Hit)
0x000000ff (Miss)
0x00000100 (Miss)
0x00001000 (Miss)
0x0000ffff (Miss)
0x0000abcd (Miss)
0x00001234 (Miss)
0x000fffff (Miss)
Hit ratio = 20%
```

```
Fully Associative Cache Simulator for Cache Line Size: 40 bytes with predefined addresses
0x00000000 (Miss)
0x00000004 (Hit)
0x00000010 (Hit)
0x000000ff (Miss)
0x00000100 (Miss)
0x00001000 (Miss)
0x0000ffff (Miss)
0x0000abcd (Miss)
0x00001234 (Miss)
0x000fffff (Miss)
Hit ratio = 20%
```

```
Fully Associative Cache Simulator for Cache Line Size: 80 bytes with predefined addresses
0x00000000 (Miss)
0x00000004 (Hit)
0x00000010 (Hit)
0x000000ff (Miss)
0x00000100 (Miss)
0x00001000 (Miss)
0x0000ffff (Miss)
0x0000abcd (Miss)
0x00001234 (Miss)
0x000fffff (Miss)
Hit ratio = 20%
```

Tracing:

Direct Mapped Cache Simulator

Cache Line Size: 16 bytes

For a 16-byte cache line, the offset is 4 bits.

1. 0x00000000: Miss (first access)
2. 0x00000004: Hit (same cache line as 0x00000000)
3. 0x00000010: Miss (different cache line)
4. 0x000000ff: Miss (different cache line)
5. 0x00000100: Miss (different cache line)
6. 0x00001000: Miss (different cache line)
7. 0x0000ffff: Miss (different cache line)
8. 0x0000abcd: Miss (different cache line)
9. 0x00001234: Miss (different cache line)
10. 0x000fffff: Miss (different cache line)

Hits: 1, Misses: 9, Hit ratio: 10%

Cache Line Size: 32 bytes

For a 32-byte cache line, the offset is 5 bits.

1. 0x00000000: Miss (first access)
2. 0x00000004: Hit (same cache line as 0x00000000)
3. 0x00000010: Hit (same cache line as 0x00000000)
4. 0x000000ff: Miss (different cache line)
5. 0x00000100: Miss (different cache line)
6. 0x00001000: Miss (different cache line)
7. 0x0000ffff: Miss (different cache line)
8. 0x0000abcd: Miss (different cache line)
9. 0x00001234: Miss (different cache line)
10. 0x000fffff: Miss (different cache line)

Hits: 2, Misses: 8, Hit ratio: 20%

Cache Line Size: 64 bytes

For a 64-byte cache line, the offset is 6 bits.

1. 0x00000000: Miss (first access)
2. 0x00000004: Hit (same cache line as 0x00000000)
3. 0x00000010: Hit (same cache line as 0x00000000)
4. 0x000000ff: Miss (different cache line)
5. 0x00000100: Miss (different cache line)
6. 0x00001000: Miss (different cache line)
7. 0x0000ffff: Miss (different cache line)
8. 0x0000abcd: Miss (different cache line)

9. 0x00001234: Miss (different cache line)
10. 0x000fffff: Miss (different cache line)

Hits: 2, Misses: 8, Hit ratio: 20%

Cache Line Size: 128 bytes

For a 128-byte cache line, the offset is 7 bits.

1. 0x00000000: Miss (first access)
2. 0x00000004: Hit (same cache line as 0x00000000)
3. 0x00000010: Hit (same cache line as 0x00000000)
4. 0x000000ff: Miss (different cache line)
5. 0x00000100: Miss (different cache line)
6. 0x00001000: Miss (different cache line)
7. 0x0000ffff: Miss (different cache line)
8. 0x0000abcd: Miss (different cache line)
9. 0x00001234: Miss (different cache line)
10. 0x000fffff: Miss (different cache line)

Hits: 2, Misses: 8, Hit ratio: 20%

Fully Associative Cache Simulator

Cache Line Size: 16 bytes

1. 0x00000000: Miss (first access)
2. 0x00000004: Hit (same cache line as 0x00000000)
3. 0x00000010: Miss (new cache line)
4. 0x000000ff: Miss (new cache line)
5. 0x00000100: Miss (new cache line)
6. 0x00001000: Miss (new cache line)
7. 0x0000ffff: Miss (new cache line)
8. 0x0000abcd: Miss (new cache line)
9. 0x00001234: Miss (new cache line)
10. 0x000fffff: Miss (new cache line)

Hits: 1, Misses: 9, Hit ratio: 10%

Cache Line Size: 32 bytes

1. 0x00000000: Miss (first access)
2. 0x00000004: Hit (same cache line as 0x00000000)
3. 0x00000010: Hit (same cache line as 0x00000000)
4. 0x000000ff: Miss (new cache line)
5. 0x00000100: Miss (new cache line)
6. 0x00001000: Miss (new cache line)
7. 0x0000ffff: Miss (new cache line)
8. 0x0000abcd: Miss (new cache line)
9. 0x00001234: Miss (new cache line)
10. 0x000fffff: Miss (new cache line)

Hits: 2, Misses: 8, Hit ratio: 20%

Cache Line Size: 64 bytes

1. 0x00000000: Miss (first access)
2. 0x00000004: Hit (same cache line as 0x00000000)
3. 0x00000010: Hit (same cache line as 0x00000000)
4. 0x000000ff: Miss (new cache line)
5. 0x00000100: Miss (new cache line)
6. 0x00001000: Miss (new cache line)
7. 0x0000ffff: Miss (new cache line)
8. 0x0000abcd: Miss (new cache line)
9. 0x00001234: Miss (new cache line)
10. 0x000fffff: Miss (new cache line)

Hits: 2, Misses: 8, Hit ratio: 20%

Cache Line Size: 128 bytes

1. 0x00000000: Miss (first access)
2. 0x00000004: Hit (same cache line as 0x00000000)
3. 0x00000010: Hit (same cache line as 0x00000000)
4. 0x000000ff: Miss (new cache line)
5. 0x00000100: Miss (new cache line)
6. 0x00001000: Miss (new cache line)
7. 0x0000ffff: Miss (new cache line)
8. 0x0000abcd: Miss (new cache line)
9. 0x00001234: Miss (new cache line)
10. 0x000fffff: Miss (new cache line)

Hits: 2, Misses: 8, Hit ratio: 20%

References:

- https://techdifferences.com/difference-between-cache-memory-and-main-memory.html#google_vignette
- CSCE-231 Lecture slides by Dr. Mohamed Shaalan

Contributions:

This project is done entirely by Jana Fadl.