M-way Set-Associative and Sector Mapped Cache in Verilog- Design and Testing

By

Kamlesh Kumar

Student ID: 017047212

Objectives

- **Develop a Cache Simulator**
  - Implemented in Verilog to analyze and compare cache performance metrics.
- **Evaluate Two Cache Architectures**:
  - **M-Way Set-Associative Cache**
    - Configurations range from direct-mapped to fully associative (32-way).
  - **Sector-Mapped Cache**
    - Introduces finer granularity by dividing cache lines into smaller sectors.
- **Analyze Key Performance Metrics**:
  - **Hit Ratio** and **Miss Ratio** across different configurations.
- **Optimize Cache Design**:
  - Explore the impact of **cache size**, **block size**, **associativity**, and **sector size**.
  - Balance performance improvement with minimal storage and hardware cost.

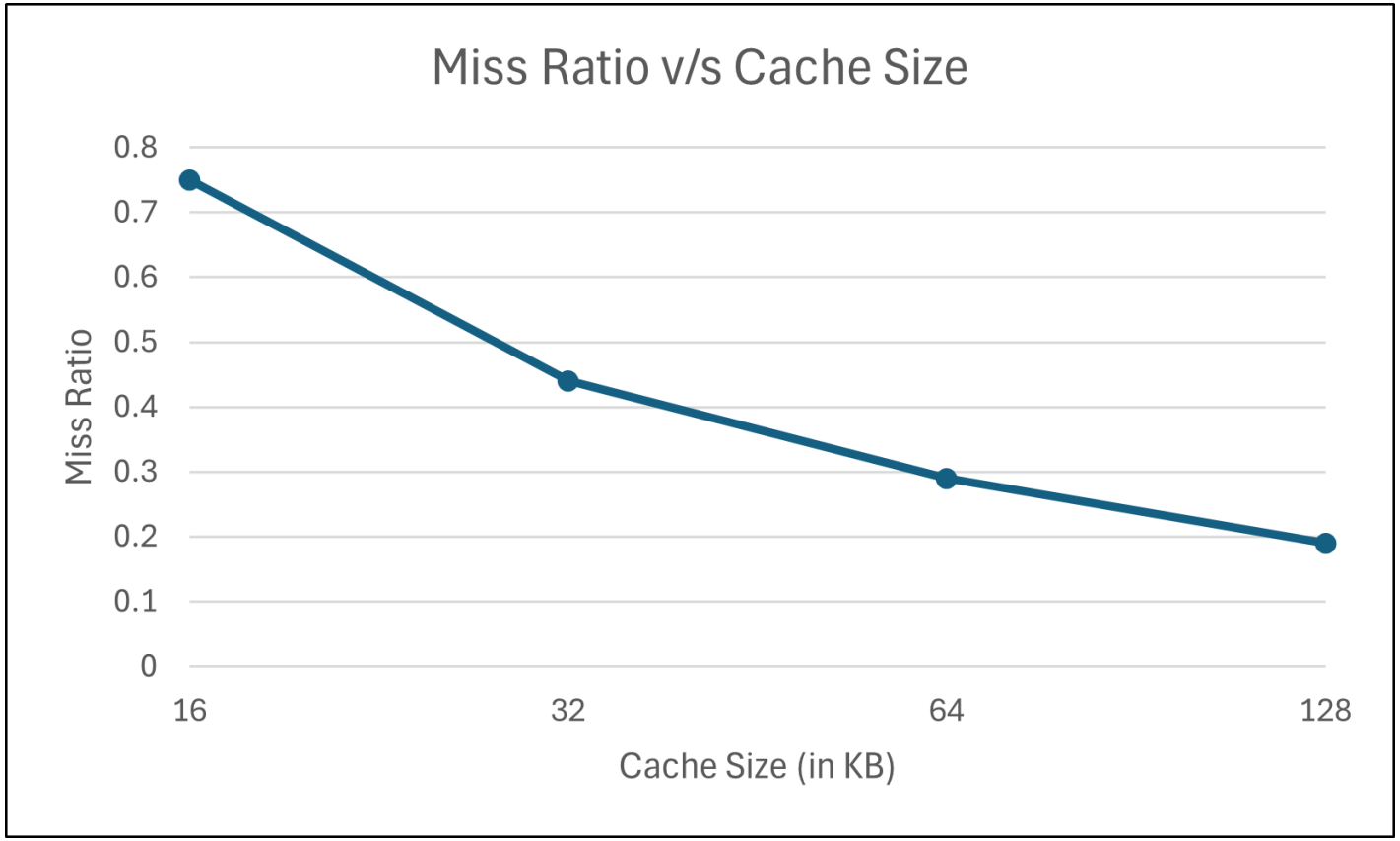
Methodology

- **Implementation**
  - Developed a **cache simulator** using:
    - **Verilog Hardware Description Language** for cache design and testing.
    - **VSCode IDE** for coding, debugging, and simulation.
  - Incorporated **configurable parameters** such as:
    - Cache size, block size, associativity, and sector size.
  - Implemented **Least Recently Used (LRU)** replacement algorithm for block management.
- **Performance Comparison**
  - Performed detailed simulations to analyze cache performance:
    - Compared **miss ratio** and **hit ratio** across varying configurations.
    - Explored the impact of:
      - Cache size (8 KB to 128 KB), Block size (8 bytes to 128 bytes), Associativity (direct-mapped to fully associative), Sector size (2 bytes to 64 bytes).
  - Collected and visualized results for performance evaluation.

M-way Set-Associative Cache: Case 1

- **Variable Cache Size**: 16, 32, 64,128-KB, while
  - Block Size = 32 bytes
  - Associativity = 4
  - Total Access Count = 1,500,000

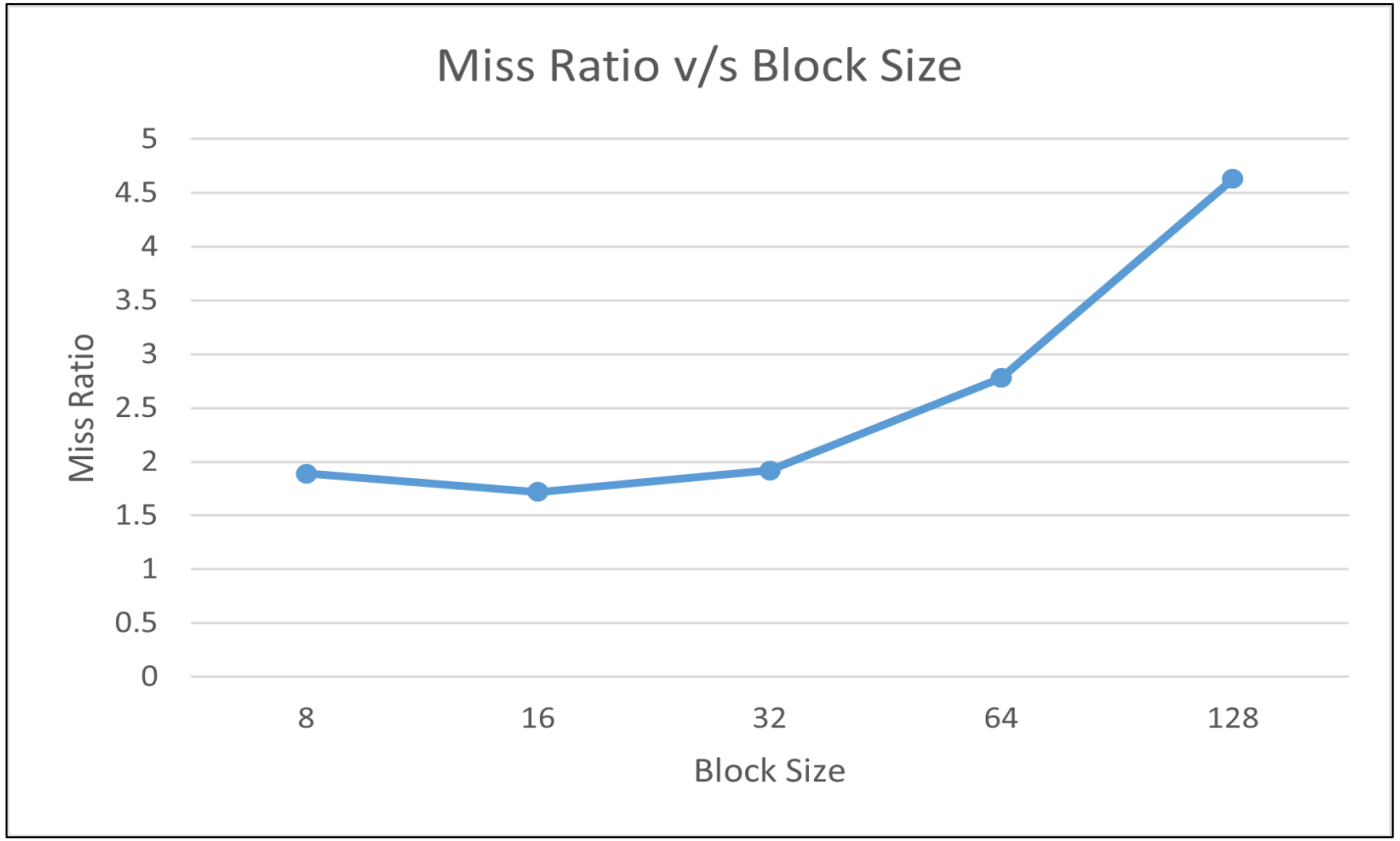
Cache Size (in KB)	%	
	Hit Ratio	Miss Ratio
16	99.25	0.75
32	99.56	0.44
64	99.71	0.29
128	99.81	0.19



Case 2: Variable *Block Size*

- **Block Size**: 16, 32, 64,128-bytes, while
  - Cache Size = 8 KB
  - Associativity = 4
  - Total Access Count = 1,500,000

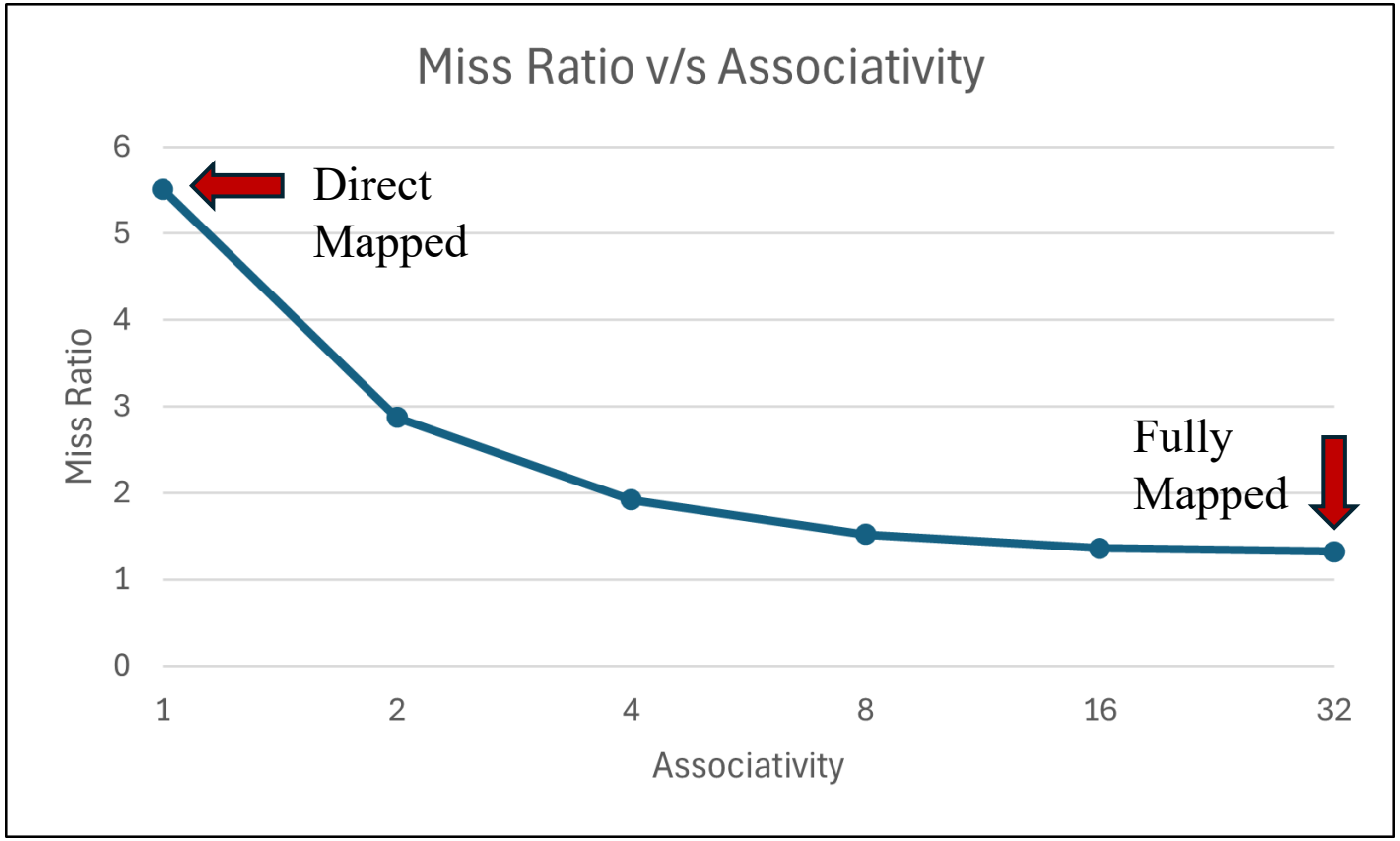
Block Size	%	
	Hit Ratio	Miss Ratio
8	98.11	1.89
16	98.28	1.72
32	98.08	1.92
64	97.28	2.78
128	95.37	4.63



Case 3: Variable *Associativity*

- **Associativity**: 1, 2, 4, 8, 16, 32-ways, while
  - Cache Size = 8 KB
  - Block Size = 32 bytes
  - Total Access Count = 1,500,000

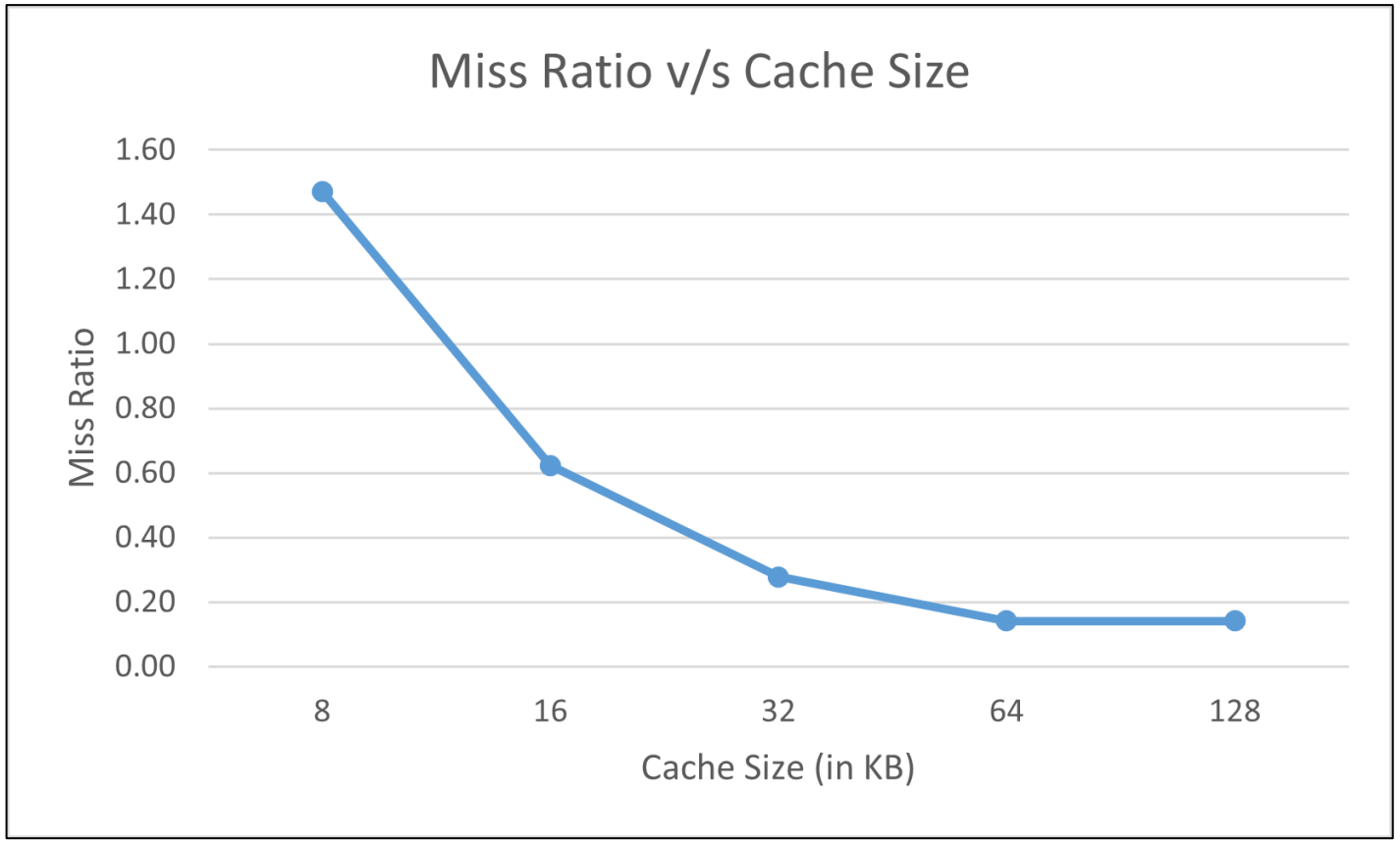
Set Associativity	%	
	Hit Ratio	Miss Ratio
1	94.49	5.51
2	97.13	2.87
4	98.08	1.92
8	98.48	1.52
16	98.64	1.36
32	98.68	1.32



Sector Mapped Cache: Case 1

- **Variable Cache Size**: 8, 16, 32, 64,128-KB, while
  - Block Size = 32 bytes
  - Associativity = 4-way
  - Sector Size = 4 bytes
  - Total Access Count = 1,500,000

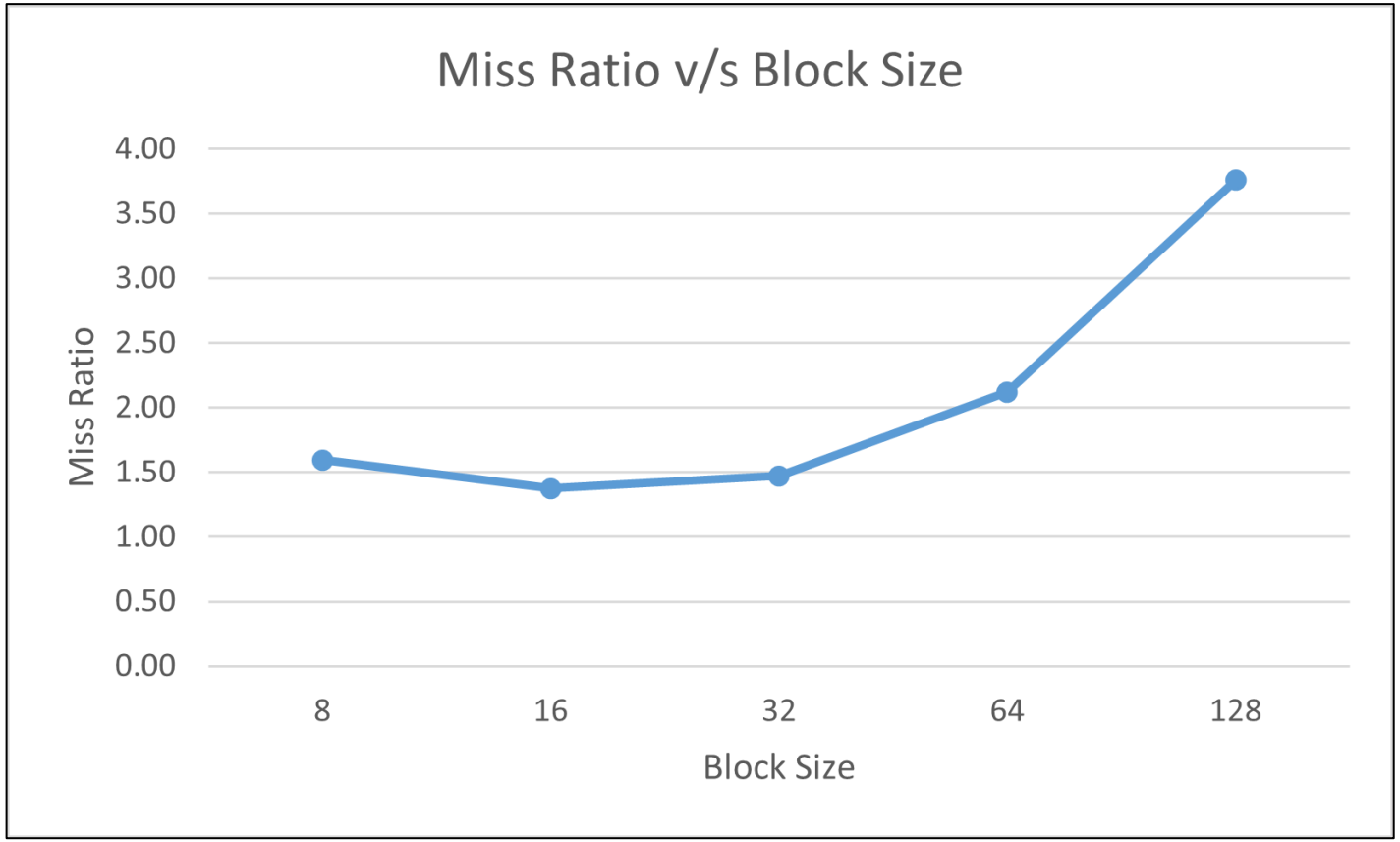
Cache Size (in KB)	%	
	Hit Ratio	Miss Ratio
8	98.53	1.47
16	99.38	0.62
32	99.72	0.28
64	99.86	0.14
128	99.86	0.14



Case 2: Variable *Block Size*

- **Block Size**: 8, 16, 32, 64, 128-bytes, while
  - Cache Size = 8 KB
  - Associativity = 4-way
  - Sector Size = 4 bytes
  - Total Access Count = 1,500,000

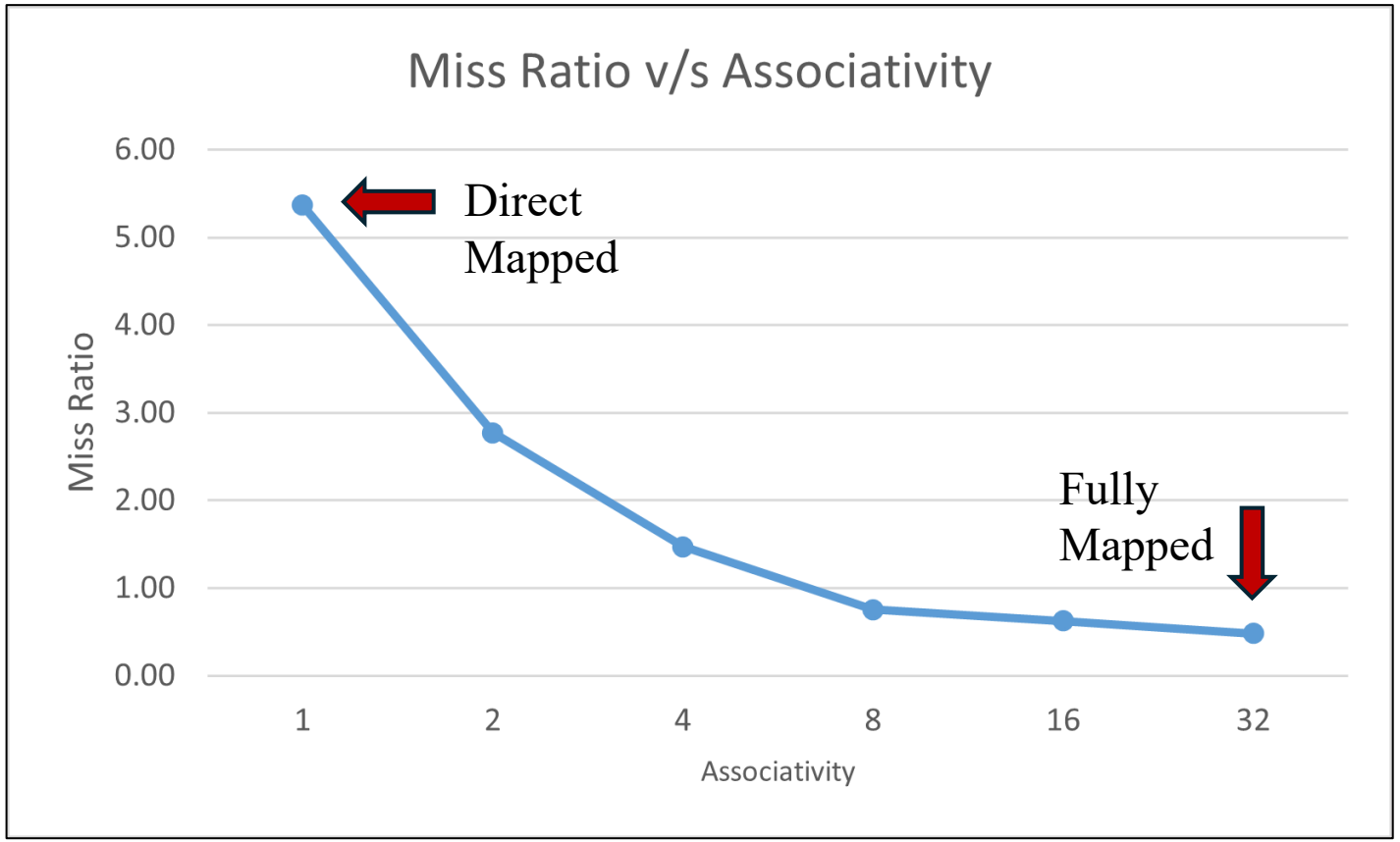
Block Size	%	
	Hit Ratio	Miss Ratio
8	98.41	1.59
16	98.62	1.38
32	98.53	1.47
64	97.88	2.12
128	96.24	3.76



Case 3: Variable *Associativity*

- **Associativity**: 1, 2, 4, 8, 16, 32-ways, while
  - Cache Size = 8 KB
  - Block Size = 32 bytes
  - Sector Size = 4 bytess
  - Total Access Count = 1,500,000

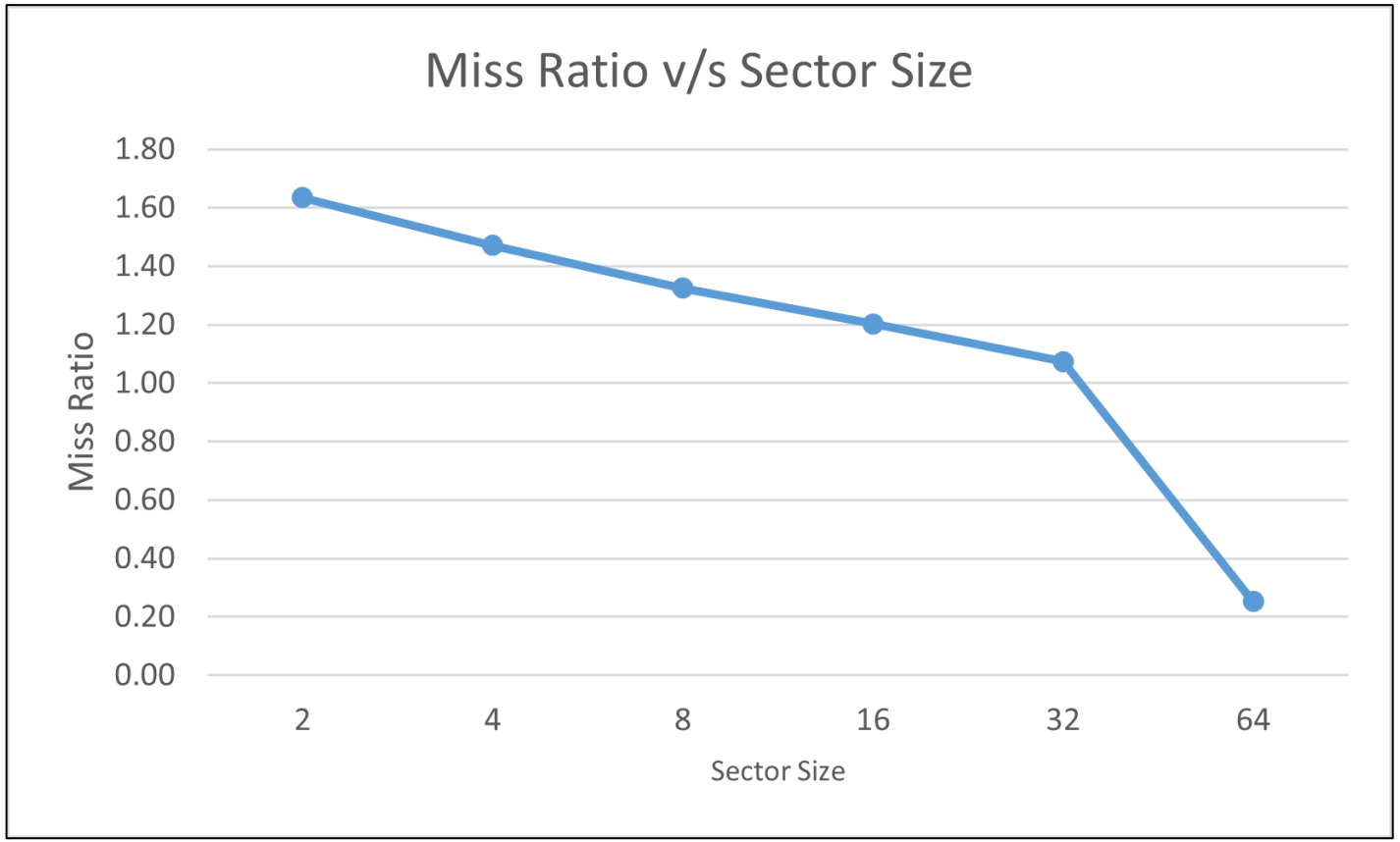
Set Associativity	%	
	Hit Ratio	Miss Ratio
1	94.63	5.37
2	97.23	2.77
4	98.53	1.47
8	99.24	0.76
16	99.38	0.62
32	99.52	0.48



Case 3: Variable *Sector Size*

- **Sector Size**: 2, 4, 8, 16, 32, 64-bytes, while
  - Cache Size = 8 KB
  - Associativity = 4-way
  - Block Size = 32 bytes
  - Total Access Count = 1,500,000

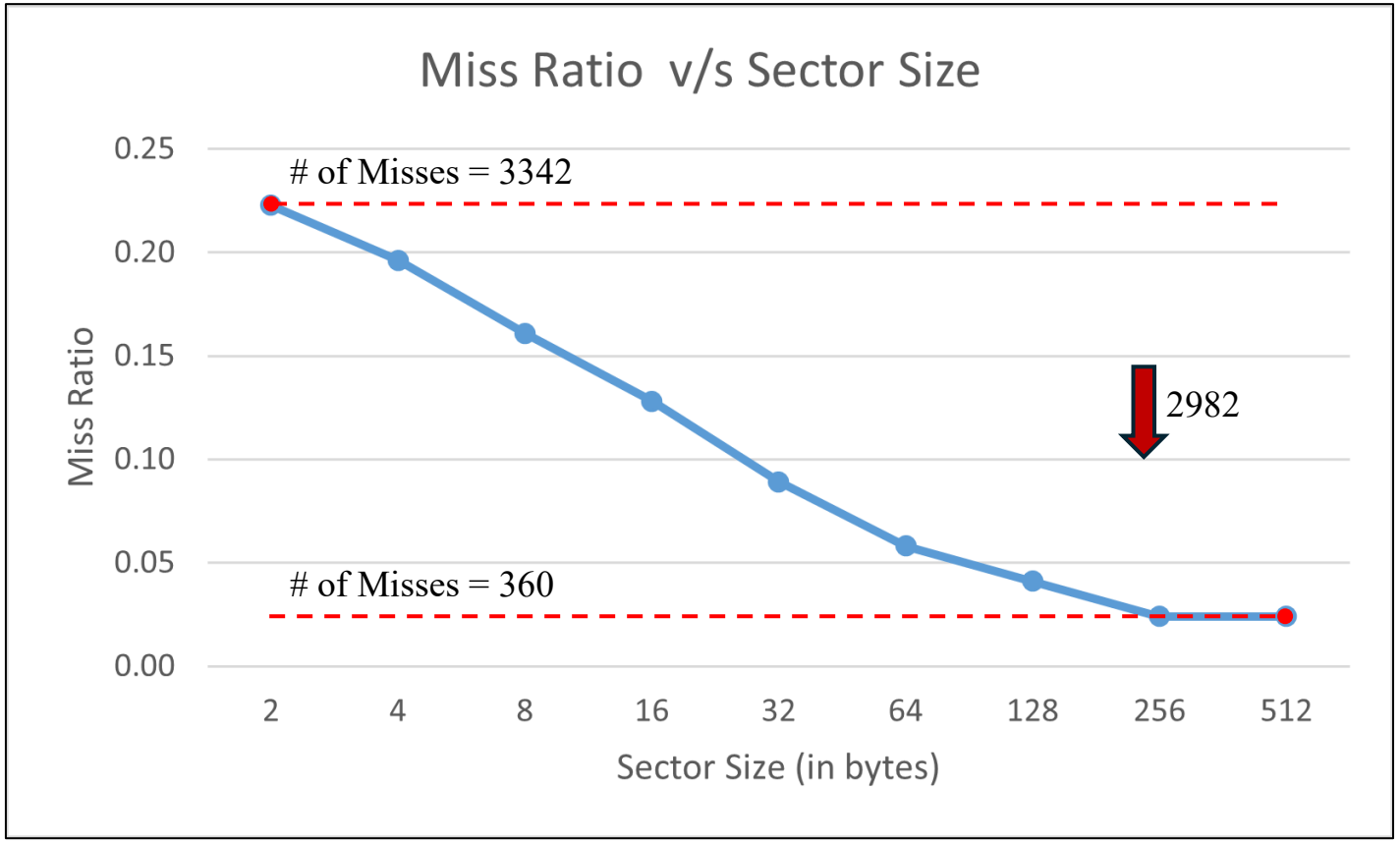
Sector Size (in bytes)	%	
	Hit Ratio	Miss Ratio
2	98.37	1.63
4	98.53	1.47
8	98.68	1.32
16	98.80	1.20
32	98.93	1.07
64	99.75	0.25



M-way v/s Sector Mapped

- **Sector Size** between 2 to 512-bytes, with **32-way Associativity** (fully-mapped), **128 KB Cache Size**, and **16 byte Block Size**.
  - Total Access Count = 1,500,000

Sector Size	Hit Ratio	Miss Ratio
2	99.78	0.22
4	99.80	0.20
8	99.84	0.16
16	99.87	0.13
32	99.91	0.09
64	99.94	0.06
128	99.96	0.04
256	**99.98**	**0.02**
512	**99.98**	**0.02**



Conclusions

- **Cache Size**:
  - **Increasing** cache size significantly **improved** performance by reducing capacity misses. For example, increasing cache size from 16 KB to 128 KB improved the hit ratio from 99.25% to 99.81%.
- **Block Size**:
  - Optimal block size is critical for balancing spatial locality and cache utilization. Larger blocks improved hit ratios by fetching more adjacent data but introduced **inefficiencies by displacing frequently accessed blocks**.
- **Associativity**:
  - Higher associativity reduced conflict misses, with **fully associative caches achieving the lowest miss ratios**. However, this came at the expense of **higher hardware complexity**.
- **Sector Size**:
  - Sector mapping introduced finer granularity, enabling efficient storage utilization and further improving hit ratios.
  - Additional results (with **128 KB Cache Size**, **16-byte Block Size**, and **32-Way Associativity**) demonstrate that increasing sector size consistently enhanced performance:
    - **Sector Size = 256 bytes: Hit Ratio = 99.98%, Miss Ratio = 0.02%.**
- **Sector Mapping for Performance Improvement**:
  - The concept of sector mapping offers significant potential for enhancing performance in workloads with irregular access patterns. By selectively caching only the relevant portions of data, it **reduces memory traffic and improves overall cache efficiency**.

Thanks!

Q & A