### Project Plan (September 16, 2024 - September 25, 2024)

#### *****Step 1: Understanding Cache Basics (Sept 16 - Sept 17, 2024)*****

**September 16, 2024:**

**Task:** Review cache types (Direct-mapped, Fully Associative, Set-Associative), block size, hit/miss concepts, replacement policies (LRU, FIFO), and write policies (Write-through, Write-back).

**Status:** Completed cache type understanding and the impact of block sizes on performance. Reviewed hit/miss rates and the importance of replacement and write policies.

**September 17, 2024:**

**Task:** Summarize findings into a brief document and research modern cache architectures used in processors.

**Status:** Finished document summarizing basic concepts of cache architectures.

#### *****Step 2: Define Cache Parameters (Sept 18, 2024)*****

**Task:** Define cache parameters (cache size, block size, associativity). Write pseudo-code or formulas for calculating cache parameters (S = C / (B \* A)).

**Status:** Parameters defined for cache design, including associativity and block size. Started pseudo-code.

\*\*\*\*Psuedo-code uploaded on Git.\*\*\*\*

#### *****Step 3: Decoding Addresses (Sept 19, 2024)*****

**Task:** Split a memory address into Tag, Index, and Offset bits. Write functions to calculate the index, tag, and offset.

**Status:** Started code implementation for decoding memory addresses.

* **Excel Entry**:
  + Date: Sept 19, 2024
  + Task: Address decoding, functions for tag, index, and offset.
  + Status: In progress.

#### *****Step 4: Data Structures (Sept 20, 2024)*****

* **Task**: Implement data structures for the cache (CacheLine, CacheSet) in C++.
* **Status**: Completed data structure implementation.
* **Excel Entry**:
  + Date: Sept 20, 2024
  + Task: Implemented cache data structures (CacheLine, CacheSet).
  + Status: Completed.

#### *****Step 5: Cache Operations (Sept 21, 2024)*****

* **Task**: Write functions for cache lookup, handling cache misses, and writing to the cache (write-through and write-back).
* **Status**: Basic lookup function implemented. Started work on handling cache misses.
* **Excel Entry**:
  + Date: Sept 21, 2024
  + Task: Cache lookup function implemented.
  + Status: In progress.

#### *****Step 6: Replacement Policy (Sept 22, 2024)*****

* **Task**: Implement Least Recently Used (LRU) replacement policy.
* **Status**: Replacement policy implementation started.
* **Excel Entry**:
  + Date: Sept 22, 2024
  + Task: Started LRU implementation.
  + Status: In progress.

#### *****Step 7: Testing Your Cache (Sept 23, 2024)*****

* **Task**: Write test cases for sequential, random, and mixed memory access patterns. Measure hit/miss rates.
* **Status**: Basic test cases written for sequential access.
* **Excel Entry**:
  + Date: Sept 23, 2024
  + Task: Created test cases for sequential access.
  + Status: In progress.

#### *****Step 8: Performance Analysis (Sept 24, 2024)*****

* **Task**: Measure performance metrics (Hit Rate, Miss Penalty, AMAT).
* **Status**: Performance metrics implementation in progress.
* **Excel Entry**:
  + Date: Sept 24, 2024
  + Task: Started calculating hit rate and miss penalty.
  + Status: In progress.

#### *****Step 9: Optimization (Sept 25, 2024)*****

* **Task**: Implement optimizations such as prefetching, victim cache, and write buffers.
* **Status**: Prefetching mechanism implemented.
* **Excel Entry**:
  + Date: Sept 25, 2024
  + Task: Prefetching implemented.
  + Status: In progress.

#### *****Step 10: Final Implementation (Sept 26, 2024)*****

* **Task**: Integrate all components and run performance tests.
* **Status**: Final integration started, preparing for performance tests.
* **Excel Entry**:
  + Date: Sept 26, 2024
  + Task: Integration and testing.
  + Status: In progress.