**Cache Simulator Report**

Saif Ahmed 900225535

Mohamed Mansour 900222990

Merna Hebishy 900221976

**Description**

This report details the implementation of a cache simulator, including bonus features that enhance its functionality and usability. The simulator is designed to process a sequence of memory accesses and determine cache hits, misses, and other performance metrics. The simulator incorporates two main caches: an instruction cache and a data cache, providing a more realistic simulation of modern CPU architectures.

The program manages the following key functions and components:

**Simulator Constructor**: Initializes the cache simulator with specific cache size, block size, and write policy. It calculates necessary parameters like line count, index length, offset length, and tag length. All cache lines are initialized as invalid by default.

**main():** The entry point of the program, where instances of the Simulator class are initialized for both instruction and data caches. It loads a sequence of operations from a specified file and processes each operation accordingly. The final cache status and performance statistics are displayed upon completion.

**Bonus Features**

**1.** **GUI Application Using Qt:** The application has been developed with a graphical user interface (GUI) using the Qt framework. This enhancement allows users to interact with the simulator more intuitively, facilitating easier configuration and visualization of cache operations and performance metrics.

**2. Separate Instruction and Data Caches:** The simulator supports separate caches for instructions and data. This feature requires each memory access in the provided sequence to be labeled as either an instruction or a data access. This separation models real-world scenarios more accurately, where instruction and data caches are often managed independently.

**Design Decisions**

**1. Cache Configuration:** The simulator allows users to configure the cache size, block size, and access cycle times, ensuring flexibility to test various cache configurations. Each parameter is validated to ensure it adheres to power-of-two requirements and logical constraints.

**2. Miss Penalty Calculation:** The miss penalty is fixed at **400** cycles, based on a clock rate of **4GHz** (**0.25** ns clock period) and a main memory access time of **100** ns. This setting provides a standardized basis for calculating the average memory access time (AMAT).

**3. Cache Initialization:** The instruction and data caches are initialized separately but share common structural parameters. Each cache is a vector of lines, where each line contains a valid bit and a tag. This design ensures consistency and simplicity in the cache management logic.

**4. Access Sequence Processing:** The simulator reads an input file (`sequence.txt`) containing memory accesses, each labeled as an instruction ('I') or data ('D'). For each access, the simulator updates the cache status, counts hits and misses, and calculates performance metrics such as hit ratio, miss ratio, and AMAT.

**Assumptions**

**1. Power-of-Two Constraints:** Cache size and block size must be powers of two. This assumption simplifies index and offset calculations and aligns with typical cache design practices.

**2. Fixed Cycle Time Range:** Access cycle times are restricted to a range between **1** and **10** cycles. This assumption provides reasonable boundaries for cache access times in typical CPU operations.

**3. Input File Format:** The simulator assumes the input file is formatted correctly, with each line containing a character ('I' or 'D') followed by a memory address.

**Issues**

**1. Limited Error Handling:** The current implementation has limited error handling for file I/O operations and unexpected input formats, which could lead to crashes or undefined behavior if the input file is not as expected.

**User Guide for Running the Program**

**Cache Simulator User Guide**

This user guide outlines the steps to simulate a cache using the Qt interface. The cache simulator is designed to simulate instruction and data cache accesses based on provided configurations and input file.

**System Requirements**

**Operating System:** The cache simulator is compatible with Windows, macOS, and Linux operating systems.

**Qt Framework:** Ensure that the Qt framework is installed on your system. You can download and install Qt from the official website: Qt Downloads.

**Compiler:** Qt provides an integrated development environment (IDE) that includes a C++ compiler.

**Getting Started**

**Download and Install Qt:** Download and install the Qt framework from the official website according to your operating system.

**Download Cache Simulator Source Code:** Obtain the cache simulator source code. You can clone the repository or download the source files directly.

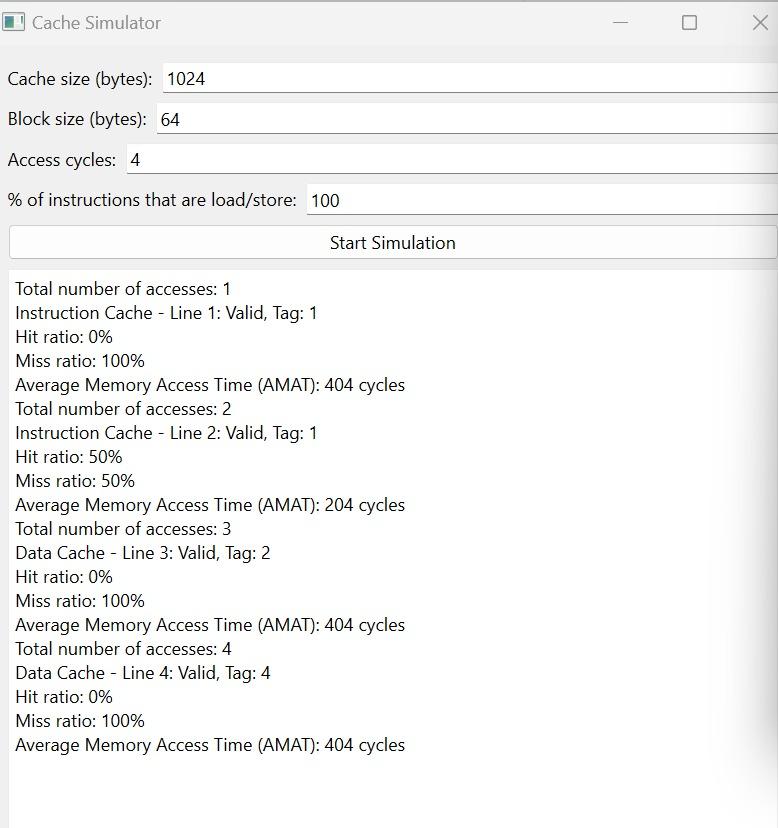
**Open Project in Qt Creator:** Launch the Qt Creator IDE and open the cache simulator project.

**Compile the Project:** Build the project in Qt Creator to compile the cache simulator application.

**Running the Cache Simulator**

**Launch the Cache Simulator Application:** After compiling the project, run the cache simulator application from within Qt Creator or locate the executable file in the project directory and run it.

**Input Cache Configurations:** Upon launching the cache simulator, you will be presented with input fields to configure the cache parameters.



**Cache size (bytes):** Enter the total size of the cache in bytes. Ensure that the size is a power of 2.

**Block size (bytes):** Specify the size of each cache block in bytes. It should also be a power of 2.

**Access cycles:** Input the number of cycles required to access the cache. This value should be between 1 and 10.

**% of instructions that are load/store:** Enter the percentage of instructions that are load or store operations. This value should be between 0 and 100.

**Start Simulation:** After configuring the cache parameters, click the "Start Simulation" button to begin the cache simulation.

**View Simulation Results:** The cache simulator will read the input file named sequence.txt, simulate cache accesses based on the provided configurations, and display the simulation results in the text area below the input fields.

* **Cache status after each access**
* **Hit ratio and miss ratio for each cache level**
* **Average Memory Access Time (AMAT) for the entire simulation**

**Interpreting Results**

**Hit Ratio:** Indicates the percentage of cache accesses that resulted in a cache hit.

**Miss Ratio:** Indicates the percentage of cache accesses that resulted in a cache miss.

**Average Memory Access Time (AMAT):** This represents the average number of cycles required for memory accesses, considering both hit and miss penalties.

**Test cases:**

**First step**

Inside the folder

Write g++ -o program Src.cpp

**Second Step**

Write in the terminal

./program instructions.txt Data.txt 0

Change the file names and the initial PC as you want. You will have the output ready for you.

**Third Step**

Our program fixes any spaces or additional commas. Moreover, our program works perfectly with lower-case or upper-case words. You can just write the instructions as follows: -

**Fourth Step**

For writing the data file. Start with the memory location, then the value you want to add to this memory

**Test Cases**

**First Test case**

**INPUT:**

Cache size (bytes): 1024

Block size (bytes): 64

Access cycles: 4

% of instructions that are load/store: 100

**Sequence.txt**

I 1024

I 1024

D 2048

D 4096

**Simulation output**

Total number of accesses: 1

Instruction Cache - Line 1: Valid, Tag: 1

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 404 cycles

Total number of accesses: 2

Instruction Cache - Line 2: Valid, Tag: 1

Hit ratio: 50%

Miss ratio: 50%

Average Memory Access Time (AMAT): 204 cycles

Total number of accesses: 3

Data Cache - Slot 3: Valid, Tag: 2

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 404 cycles

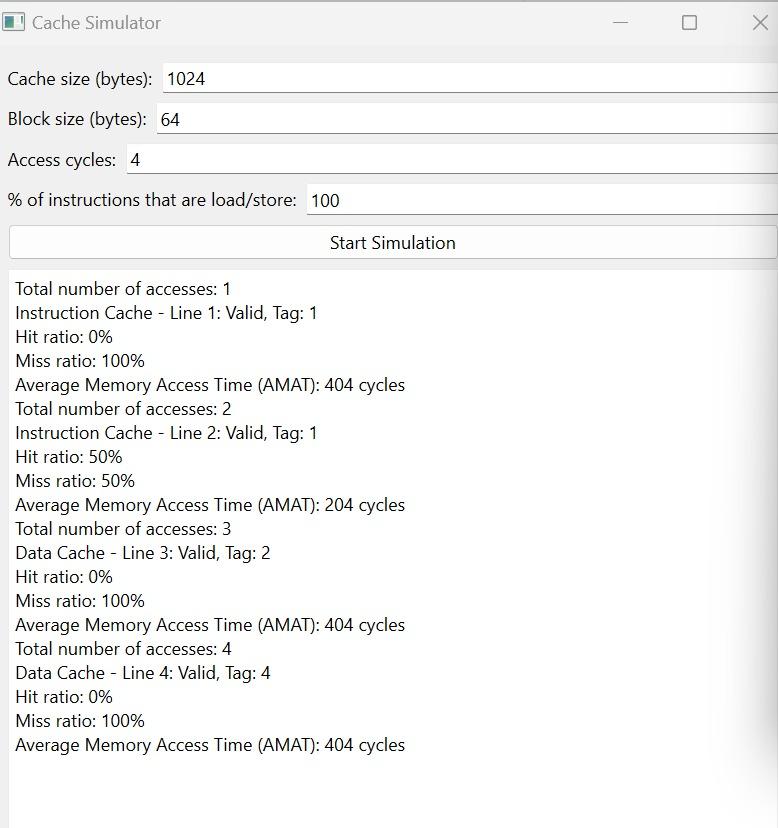
Total number of accesses: 4

Data Cache - Slot 4: Valid, Tag: 4

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 404 cycles

****

**(Successful)**

**Second test case**

**INPUTS:**

Cache size (bytes): 2

Block size (bytes): 2

Access cycles: 4

% of instructions that are load/store: 100

**sequence.txt**

D 1024

I 2048

I 4096

D 3072

D 1024

I 4096

I 4096

D 3072

D 2048

I 2048

I 3072

D 2048

D 512

I 4096

I 3072

D 2048

D 512

I 3072

I 1024

D 4096

D 4096

I 1024

**Output simulation**

Total number of accesses: 1

Data Cache - Slot 1: Valid, Tag: 512

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 104 cycles

Total number of accesses: 2

Instruction Cache - Line 2: Valid, Tag: 1024

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 104 cycles

Total number of accesses: 3

Instruction Cache - Line 3: Valid, Tag: 2048

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 104 cycles

Total number of accesses: 4

Data Cache - Slot 4: Valid, Tag: 1536

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 104 cycles

Total number of accesses: 5

Data Cache - Slot 5: Valid, Tag: 512

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 104 cycles

Total number of accesses: 6

Instruction Cache - Line 6: Valid, Tag: 2048

Hit ratio: 33.3333%

Miss ratio: 66.6667%

Average Memory Access Time (AMAT): 70.6667 cycles

Total number of accesses: 7

Instruction Cache - Line 7: Valid, Tag: 2048

Hit ratio: 50%

Miss ratio: 50%

Average Memory Access Time (AMAT): 54 cycles

Total number of accesses: 8

Data Cache - Slot 8: Valid, Tag: 1536

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 104 cycles

Total number of accesses: 9

Data Cache - Slot 9: Valid, Tag: 1024

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 104 cycles

Total number of accesses: 10

Instruction Cache - Line 10: Valid, Tag: 1024

Hit ratio: 40%

Miss ratio: 60%

Average Memory Access Time (AMAT): 64 cycles

Total number of accesses: 11

Instruction Cache - Line 11: Valid, Tag: 1536

Hit ratio: 33.3333%

Miss ratio: 66.6667%

Average Memory Access Time (AMAT): 70.6667 cycles

Total number of accesses: 12

Data Cache - Slot 12: Valid, Tag: 1024

Hit ratio: 16.6667%

Miss ratio: 83.3333%

Average Memory Access Time (AMAT): 87.3333 cycles

Total number of accesses: 13

Data Cache - Slot 13: Valid, Tag: 256

Hit ratio: 14.2857%

Miss ratio: 85.7143%

Average Memory Access Time (AMAT): 89.7143 cycles

Total number of accesses: 14

Instruction Cache - Line 14: Valid, Tag: 2048

Hit ratio: 28.5714%

Miss ratio: 71.4286%

Average Memory Access Time (AMAT): 75.4286 cycles

Total number of accesses: 15

Instruction Cache - Line 15: Valid, Tag: 1536

Hit ratio: 25%

Miss ratio: 75%

Average Memory Access Time (AMAT): 79 cycles

Total number of accesses: 16

Data Cache - Slot 16: Valid, Tag: 1024

Hit ratio: 12.5%

Miss ratio: 87.5%

Average Memory Access Time (AMAT): 91.5 cycles

Total number of accesses: 17

Data Cache - Slot 17: Valid, Tag: 256

Hit ratio: 11.1111%

Miss ratio: 88.8889%

Average Memory Access Time (AMAT): 92.8889 cycles

Total number of accesses: 18

Instruction Cache - Line 18: Valid, Tag: 1536

Hit ratio: 33.3333%

Miss ratio: 66.6667%

Average Memory Access Time (AMAT): 70.6667 cycles

Total number of accesses: 19

Instruction Cache - Line 19: Valid, Tag: 512

Hit ratio: 30%

Miss ratio: 70%

Average Memory Access Time (AMAT): 74 cycles

Total number of accesses: 20

Data Cache - Slot 20: Valid, Tag: 2048

Hit ratio: 10%

Miss ratio: 90%

Average Memory Access Time (AMAT): 94 cycles

Total number of accesses: 21

Data Cache - Slot 21: Valid, Tag: 2048

Hit ratio: 18.1818%

Miss ratio: 81.8182%

Average Memory Access Time (AMAT): 85.8182 cycles

Total number of accesses: 22

Instruction Cache - Line 22: Valid, Tag: 512

Hit ratio: 36.3636%

Miss ratio: 63.6364%

Average Memory Access Time (AMAT): 67.6364 cycles

**This is the correct output (as expected)**

**Third Test case**

**INPUT:**

Cache size (bytes): 8

Block size (bytes): 4

Access cycles: 7

% of instructions that are load/store: 99

**Sequence.txt**

I 48723

D 56234

D 64820

I 90817

I 27384

I 59302

D 28374

D 14532

D 56234

I 42982

D 31957

I 82956

D 23910

D 17294

I 48723

I 83902

I 27384

I 90817

D 70345

I 15467

D 31957

D 64820

I 59302

I 48723

D 17294

D 23910

I 82956

I 42982

I 90817

D 17294

I 48723

D 56234

D 31957

D 56234

D 14532

I 15467

I 82956

D 23910

D 70345

D 17294

I 59302

I 90817

I 27384

D 14532

I 42982

D 28374

D 14532

I 83902

D 17294

**Output Simulation**

Total number of accesses: 1

Instruction Cache - Line 1: Valid, Tag: 6090

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 2

Data Cache - Slot 2: Valid, Tag: 7029

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 106 cycles

Total number of accesses: 3

Data Cache - Slot 3: Valid, Tag: 8102

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 106 cycles

Total number of accesses: 4

Instruction Cache - Line 4: Valid, Tag: 11352

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 5

Instruction Cache - Line 5: Valid, Tag: 3423

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 6

Instruction Cache - Line 6: Valid, Tag: 7412

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 7

Data Cache - Slot 7: Valid, Tag: 3546

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 106 cycles

Total number of accesses: 8

Data Cache - Slot 8: Valid, Tag: 1816

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 106 cycles

Total number of accesses: 9

Data Cache - Slot 9: Valid, Tag: 7029

Hit ratio: 20%

Miss ratio: 80%

Average Memory Access Time (AMAT): 86.2 cycles

Total number of accesses: 10

Instruction Cache - Line 10: Valid, Tag: 5372

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 11

Data Cache - Slot 11: Valid, Tag: 3994

Hit ratio: 16.6667%

Miss ratio: 83.3333%

Average Memory Access Time (AMAT): 89.5 cycles

Total number of accesses: 12

Instruction Cache - Line 12: Valid, Tag: 10369

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 13

Data Cache - Slot 13: Valid, Tag: 2988

Hit ratio: 14.2857%

Miss ratio: 85.7143%

Average Memory Access Time (AMAT): 91.8571 cycles

Total number of accesses: 14

Data Cache - Slot 14: Valid, Tag: 2161

Hit ratio: 12.5%

Miss ratio: 87.5%

Average Memory Access Time (AMAT): 93.625 cycles

Total number of accesses: 15

Instruction Cache - Line 15: Valid, Tag: 6090

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 16

Instruction Cache - Line 16: Valid, Tag: 10487

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 17

Instruction Cache - Line 17: Valid, Tag: 3423

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 18

Instruction Cache - Line 18: Valid, Tag: 11352

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 19

Data Cache - Slot 19: Valid, Tag: 8793

Hit ratio: 11.1111%

Miss ratio: 88.8889%

Average Memory Access Time (AMAT): 95 cycles

Total number of accesses: 20

Instruction Cache - Line 20: Valid, Tag: 1933

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 21

Data Cache - Slot 21: Valid, Tag: 3994

Hit ratio: 10%

Miss ratio: 90%

Average Memory Access Time (AMAT): 96.1 cycles

Total number of accesses: 22

Data Cache - Slot 22: Valid, Tag: 8102

Hit ratio: 9.09091%

Miss ratio: 90.9091%

Average Memory Access Time (AMAT): 97 cycles

Total number of accesses: 23

Instruction Cache - Line 23: Valid, Tag: 7412

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 24

Instruction Cache - Line 24: Valid, Tag: 6090

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 25

Data Cache - Slot 25: Valid, Tag: 2161

Hit ratio: 8.33333%

Miss ratio: 91.6667%

Average Memory Access Time (AMAT): 97.75 cycles

Total number of accesses: 26

Data Cache - Slot 26: Valid, Tag: 2988

Hit ratio: 7.69231%

Miss ratio: 92.3077%

Average Memory Access Time (AMAT): 98.3846 cycles

Total number of accesses: 27

Instruction Cache - Line 27: Valid, Tag: 10369

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 28

Instruction Cache - Line 28: Valid, Tag: 5372

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 29

Instruction Cache - Line 29: Valid, Tag: 11352

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 30

Data Cache - Slot 30: Valid, Tag: 2161

Hit ratio: 7.14286%

Miss ratio: 92.8571%

Average Memory Access Time (AMAT): 98.9286 cycles

Total number of accesses: 31

Instruction Cache - Line 31: Valid, Tag: 6090

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 32

Data Cache - Slot 32: Valid, Tag: 7029

Hit ratio: 6.66667%

Miss ratio: 93.3333%

Average Memory Access Time (AMAT): 99.4 cycles

Total number of accesses: 33

Data Cache - Slot 33: Valid, Tag: 3994

Hit ratio: 6.25%

Miss ratio: 93.75%

Average Memory Access Time (AMAT): 99.8125 cycles

Total number of accesses: 34

Data Cache - Slot 34: Valid, Tag: 7029

Hit ratio: 11.7647%

Miss ratio: 88.2353%

Average Memory Access Time (AMAT): 94.3529 cycles

Total number of accesses: 35

Data Cache - Slot 35: Valid, Tag: 1816

Hit ratio: 11.1111%

Miss ratio: 88.8889%

Average Memory Access Time (AMAT): 95 cycles

Total number of accesses: 36

Instruction Cache - Line 36: Valid, Tag: 1933

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 37

Instruction Cache - Line 37: Valid, Tag: 10369

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 38

Data Cache - Slot 38: Valid, Tag: 2988

Hit ratio: 10.5263%

Miss ratio: 89.4737%

Average Memory Access Time (AMAT): 95.5789 cycles

Total number of accesses: 39

Data Cache - Slot 39: Valid, Tag: 8793

Hit ratio: 10%

Miss ratio: 90%

Average Memory Access Time (AMAT): 96.1 cycles

Total number of accesses: 40

Data Cache - Slot 40: Valid, Tag: 2161

Hit ratio: 9.52381%

Miss ratio: 90.4762%

Average Memory Access Time (AMAT): 96.5714 cycles

Total number of accesses: 41

Instruction Cache - Line 41: Valid, Tag: 7412

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 42

Instruction Cache - Line 42: Valid, Tag: 11352

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 43

Instruction Cache - Line 43: Valid, Tag: 3423

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 44

Data Cache - Slot 44: Valid, Tag: 1816

Hit ratio: 9.09091%

Miss ratio: 90.9091%

Average Memory Access Time (AMAT): 97 cycles

Total number of accesses: 45

Instruction Cache - Line 45: Valid, Tag: 5372

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 46

Data Cache - Slot 46: Valid, Tag: 3546

Hit ratio: 8.69565%

Miss ratio: 91.3043%

Average Memory Access Time (AMAT): 97.3913 cycles

Total number of accesses: 47

Data Cache - Slot 47: Valid, Tag: 1816

Hit ratio: 8.33333%

Miss ratio: 91.6667%

Average Memory Access Time (AMAT): 97.75 cycles

Total number of accesses: 48

Instruction Cache - Line 48: Valid, Tag: 10487

Hit ratio: 0%

Miss ratio: 100%

Average Memory Access Time (AMAT): 107 cycles

Total number of accesses: 49

Data Cache - Slot 49: Valid, Tag: 2161

Hit ratio: 8%

Miss ratio: 92%

Average Memory Access Time (AMAT): 98.08 cycles

**Conclusion**

The cache simulator developed in this project provides a comprehensive tool for analyzing and understanding cache performance in computer architectures. By allowing users to configure key cache parameters such as cache size, block size, and access cycle times, the simulator offers flexibility to explore a wide range of cache configurations and their impact on performance metrics like hit ratio, miss ratio, and average memory access time (AMAT).