Cache simulator

Test Summary Report

Version 3.0

05/04/2020

VERSION HISTORY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** |
| 3.0 | Team 03 | 05/03/2020 | Team 03 | 05/03/2020 |
| 2.0 | Team 03 | 04/27/2020 | Team 03 | 04/27/2020 |
| 1.0 | Team 03 | 04/06/2020 | Team 03 | 04/06/2020 |

TABLE OF CONTENTS

[1 Introduction 5](#_Toc216837569)

[1.1 Purpose 5](#_Toc216837570)

[2 Test Summary 5](#_Toc216837571) Version 1.0

[2.1 Test Parameters -f Trace1.trc -s 1024 -b 4 -a 1 -r RR 5](#_Toc216837572)

[2.2 Test Parameters -f Trace1.trc -s 8192 -b 4 -a 16 -r RR 5](#_Toc216837573)

[2.3 Test Parameters -f Trace1.trc -s 4096 -b 8 -a 2 -r RR 5](#_Toc216837574)

[3 Test Summary 5](#_Toc216837571) Version 2.0

[3.1 5](#_Toc216837572)Test Run 1

[3.2 Test Run 5](#_Toc216837573)2

[3.3 Test Run 3 5](#_Toc216837574)

[3.4 Test Run 5](#_Toc216837574)4

[3.5 Test Run 5](#_Toc216837574)5

[4 Test Summary 5](#_Toc216837571) version 3.0

[4.1 *Test*  5](#_Toc216837572)Parameters

[5 Test Assessment 5](#_Toc216837575)

[6 Test Results 5](#_Toc216837576)

[7 SuggEstED aCTIONS 6](#_Toc216837577)

[APPENDIX B: 8](#_Toc216837579)KEY TERMS

[APPENDIX C: 9](#_Toc216837580)Formulas for calculations

# 

# Introduction

## Purpose

This Cache Simulator Test Report provides a summary of the results of test performed as outlined within this document. It will go through details of each version of the Cache Simulator application, the parameters used for each simulation, and the results of the simulations. Lastly it will go over different options of how to configure the cache settings, cache size, block size, associativity and replacement method, for the Central Processing Unit (CPU) describing the advantages and disadvantages of the different configurations.

# Test Summary Version 1.0

**Project Name**: Cache Simulator

**Version Number**: 1.0

**Additional Comments**: Initial version of the simulator, this version was designed to take input parameters, the name of the trace file, the size of the cache (in KB), the size of each block in the cache (in bytes), the associativity of the cache, and the replacement method. Based on those parameters version 1.0 of the simulator would calculate the total number of blocks, the tag and index sizes in bits, the total number of rows, the overhead size in bytes, the implementation memory size in KB, and the cost of the cache ($0.05 / KB). After the values were calculated it would then print out the first 20 addresses found in the trace file and the length of the read from that address to ensure that the simulator was able to properly read the trace file. The addresses are printed in hex format and the length of the read is in decimal format, if the address was 0x00000000 it was ignored. This version was not done to generate results on what configuration to use for the new CPU but to ensure that the calculations were working correctly.

Command parameters is as follows

1. -f <trace file name> [ name of the text file with the trace ]
2. -s <cache size in KB> [ 1 KB to 8 MB ]
3. -b <block size> [ 4 bytes to 64 bytes ]
4. -a <associativity [ 1, 2, 4, 8, 16 ]
5. -r <replacement policy> [ RR or RND or LRU ]

## Test Parameters -f Trace1.trc -s 1024 -b 4 -a 1 -r RR

**Test Owner**: Team 03

**Test Date**: 04/05/2020

**Test Results**: 1024 KB cache size, 4 byte block size, 1-way associativity, round robin replacement method

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Total Block Count** | **Tag Size** | **Index Size** | **Total Number of Rows** | **Overhead Size** | **Implementation Memory Size** | **Cost** |
| 262144 | 12 bits | 18 bits | 262144 | 425984.0 Bytes | 1440.0 KB | $72.00 |

**Additional Comments**: Reading trace file was successful and printed first 20 addresses with length of the read at that address, ignoring any 0x00000000 addresses

## Test Parameters -f Trace1.trc -s 8192 -b 4 -a 16 -r RR

**Test Owner**: Team 03

**Test Date**: 04/05/2020

**Test Results**: 8192 KB cache size, 4 byte block size, 16-way associativity, round robin replacement method

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Total Block Count** | **Tag Size** | **Index Size** | **Total Number of Rows** | **Overhead Size** | **Implementation Memory Size** | **Cost** |
| 2097152 | 13 bits | 17 bits | 131072 | 3670016.0 Bytes | 11776.0 KB | $588.80 |

**Additional Comments**: Reading trace file was successful and printed first 20 addresses with length of the read at that address, ignoring any 0x00000000 addresses

## Test Parameters -f Trace1.trc -s 4096 -b 8 -a 2 -r RR

**Test Owner**: Team 03

**Test Date**: 04/05/2020

**Test Results**: 4096 KB cache size, 8 byte block size, 2-way associativity, round robin replacement method

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Total Block Count** | **Tag Size** | **Index Size** | **Total Number of Rows** | **Overhead Size** | **Implementation Memory Size** | **Cost** |
| 524288 | 11 bits | 18 bits | 262144 | 786432.0 Bytes | 4864.0 KB | $243.20 |

**Additional Comments**: Reading trace file was successful and printed first 20 addresses with length of the read at that address, ignoring any 0x00000000 addresses

# Test Summary Version 2.0

**Project Name**: Cache Simulator

**Version Number**: 2.0

**Additional Comments**: Second version of the cache simulator. This version was designed to run the simulations of the cache and be able to test how well the different configurations would perform. For the test done in this version the same parameters were used each time but ran a total of 5 times to see how well the simulator did at calculating the total number of accesses, number of hits, number of misses, hit rate, miss rate, CPI, and unused cache space. This version still did the same calculations as in version 1.0 as well. This version was also not meant to determine the configuration that would be used for the final assessment of the best configuration.

Command parameters is as follows

1. -f <trace file name> [ name of the text file with the trace ]
2. -s <cache size in KB> [ 1 KB to 8 MB ]
3. -b <block size> [ 4 bytes to 64 bytes ]
4. -a <associativity [ 1, 2, 4, 8, 16 ]
5. -r <replacement policy> [ RR or RND or LRU ]

Test Parameters:

-f A-9\_new\_1.5.pdf.trc -s 512 -b 16 -a 8 -r RND

512 KB cache size, 16 byte block size, 8-way associativity, random replacement method

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Total Block Count** | **Tag Size** | **Index Size** | **Total Number of Rows** | **Overhead Size** | **Implementation Memory Size** | **Cost** |
| 32768 | 16 bits | 12 bits | 4096 | 69632.0 Bytes | 580.0 KB | $29.00 |

## Test run 1

**Test Owner**: Team 03

**Test Date**: 04/27/2020

**Test Results**:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total Cache Accesses** | **Cache Hits** | **Cache Misses** | **Compulsory Misses** | **Conflict Misses** | **Hit Rate (%)** | **Miss Rate (%)** | **CPI** | **Unused Cache Space** | **Waste** | **Unused Cache Blocks** |
| 350466 | 338129 | 12337 | 12271 | 66 | 96.4798% | 3.5202% | 2.91 | 362.80 / 580.00 KB | $18.14 | 20563 / 32768 |

## Test run 2

**Test Owner**: Team 03

**Test Date**: 04/05/2020

**Test Results**:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total Cache Accesses** | **Cache Hits** | **Cache Misses** | **Compulsory Misses** | **Conflict Misses** | **Hit Rate (%)** | **Miss Rate (%)** | **CPI** | **Unused Cache Space** | **Waste** | **Unused Cache Blocks** |
| 350466 | 338123 | 12343 | 12271 | 72 | 96.4781% | 3.5219% | 2.92 | 362.80 / 580.00 KB | $18.14 | 20569 / 32768 |

## Test run 3

**Test Owner**: Team 03

**Test Date**: 04/05/2020

**Test Results**:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total Cache Accesses** | **Cache Hits** | **Cache Misses** | **Compulsory Misses** | **Conflict Misses** | **Hit Rate (%)** | **Miss Rate (%)** | **CPI** | **Unused Cache Space** | **Waste** | **Unused Cache Blocks** |
| 350466 | 338129 | 12343 | 12271 | 72 | 96.4781% | 3.5219% | 2.92 | 362.80 / 580.00 KB | $18.14 | 20569 / 32768 |

## Test run 4

**Test Owner**: Team 03

**Test Date**: 04/05/2020

**Test Results**:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total Cache Accesses** | **Cache Hits** | **Cache Misses** | **Compulsory Misses** | **Conflict Misses** | **Hit Rate (%)** | **Miss Rate (%)** | **CPI** | **Unused Cache Space** | **Waste** | **Unused Cache Blocks** |
| 350466 | 338129 | 12337 | 12271 | 66 | 96.4798% | 3.5202% | 2.91 | 362.80 / 580.00 KB | $18.14 | 20563 / 32768 |

## Test run 5

**Test Owner**: Team 03

**Test Date**: 04/05/2020

**Test Results**:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total Cache Accesses** | **Cache Hits** | **Cache Misses** | **Compulsory Misses** | **Conflict Misses** | **Hit Rate (%)** | **Miss Rate (%)** | **CPI** | **Unused Cache Space** | **Waste** | **Unused Cache Blocks** |
| 350466 | 338121 | 12271 | 12271 | 74 | 96.4775% | 3.5225% | 2.92 | 362.80 / 580.00 KB | $18.14 | 20571 / 32768 |

# Test Summary Version 3.0

**Project Name**: Cache Simulator

**Version Number**: 3.0

**Additional Comments**: This is the final version of the cache simulator, this version automated simulating each configuration of the cache sizes, block sizes, associativity, and replacement methods. The simulation was ran against two trace files, A-9\_new\_1.5.pdf.trc and Trace2A.trc, with each configuration. This simulation is meant to determine the most viable options for the configuration of the cache weighing different factors as hit rate, miss rate, cost, waste, and CPI. A total of 240 simulations were ran and the results were inserted into a csv file. In the assessment the suggestion of the most valuable configurations will be presented and explained

## Test Parameters

**Test Owner**: Team 03

**Test Date**: 05/03/2020

**Test Results**: See FinalTrace.csv

**Additional Comments:**

Parameters used for running the simulations. Each possible configuration of the values below was used.

<trace file name> [ A-9\_new\_1.5.pdf.trc, Trace2A.trc ]

<cache size in KB> [ 8 KB, 64 KB, 256 KB, 1024 KB ]

<block size> [ 4 bytes, 16 bytes, 64 bytes ]

<associativity [ 1, 2, 4, 8, 16 ]

<replacement policy> [ RR, RND ]

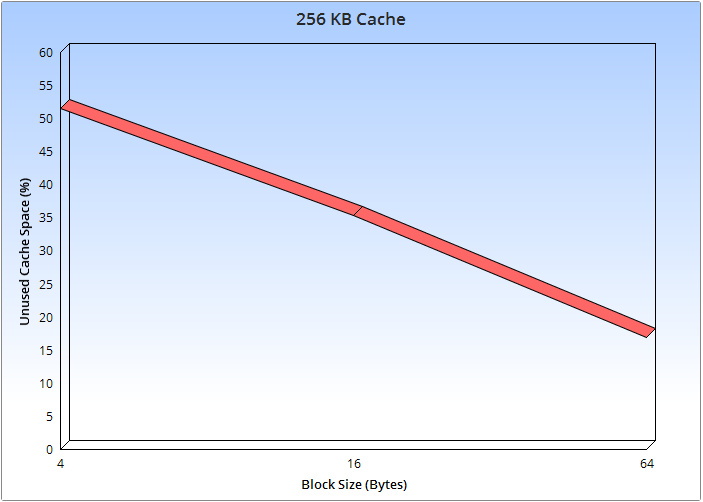
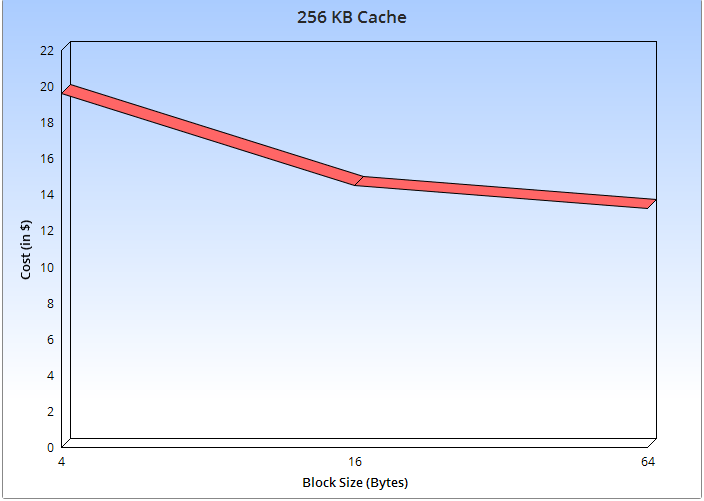
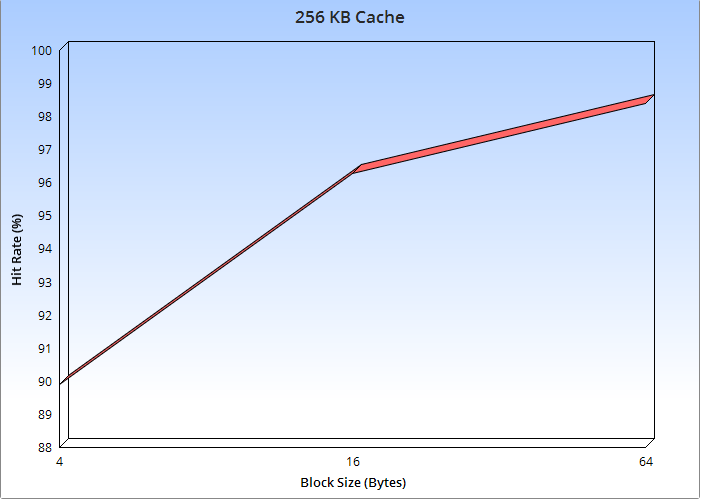
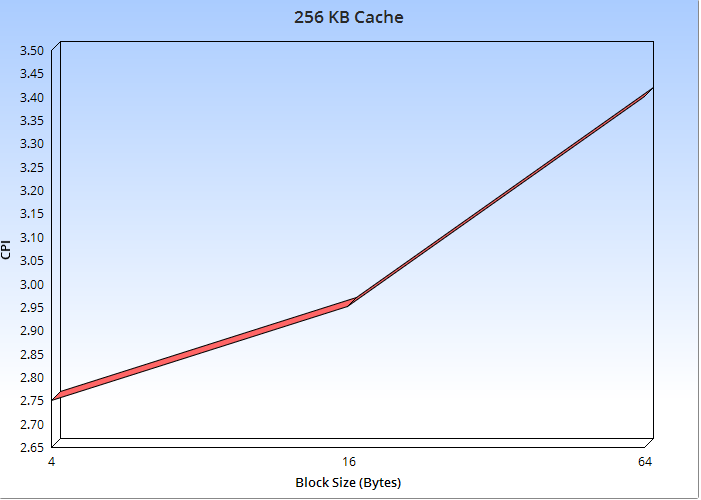
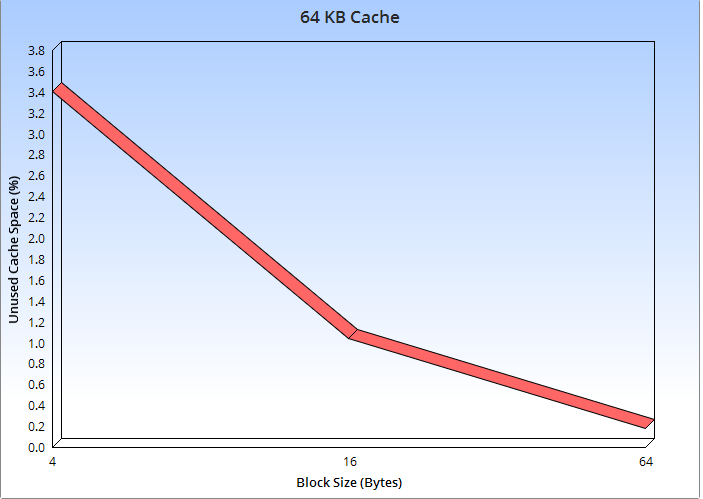
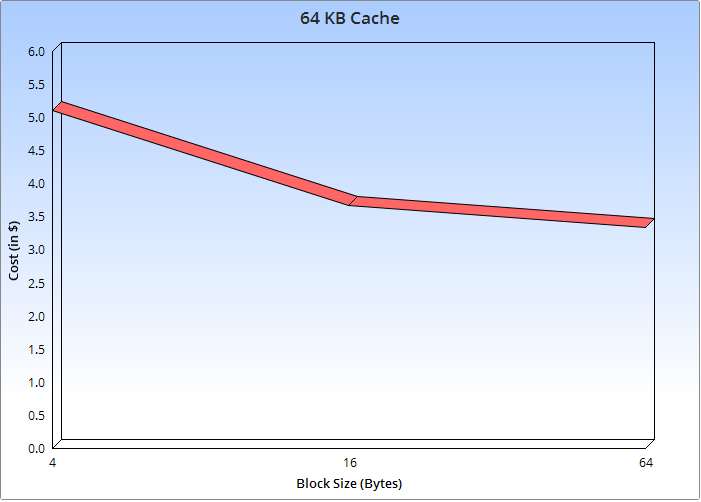
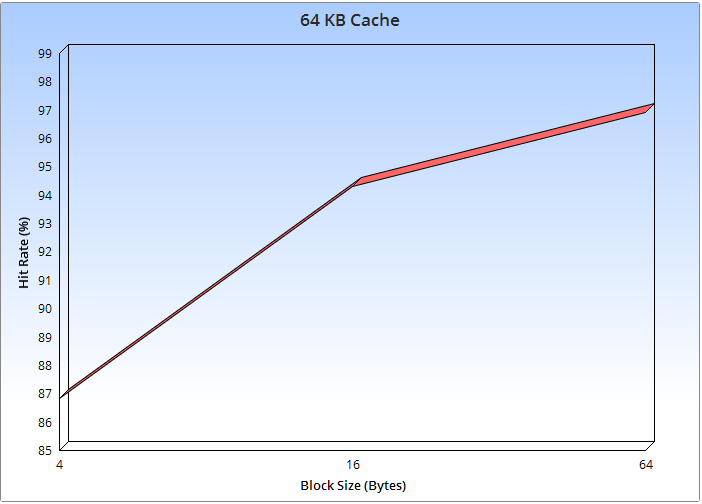
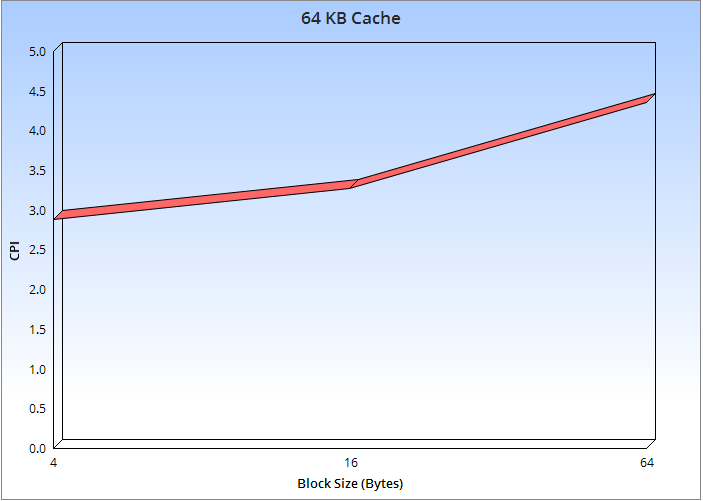
# Test Assessment

The test proved to be very successful and help in being able to determine multiple options of the best configuration to use for the cache based on cost, waste, and performance. With smaller cache sizes, 8 KB, we would have very little to zero waste in cache space but would have several conflict misses, on average 10%+ miss rate, which would cause very poor performance as there would have to be several reads to memory instead of cache. The number of reads would overall lead to a bad user experience and would not be recommended. On the other hand, if we go to the largest cache size, 1 MB, we have a hit rate of 90% to 99% but notice a large amount of unused cache space and cache blocks. We see results of very high unused cache spaces of 80% and higher which leads to a lot of wasted cache and overall cost ranging from $45.00 to $74.00 per unit. This would also not be an optimal configuration for our product. Where we saw some of the best results were in the 64 KB and 256 KB cache sizes, which we will go over multiple configurations with those cache sizes comparing differences in block size, associativity, and replacement method. The random replacement method would have the best results as for every simulation as it would be able to quickly select a block to replace versus the round robin method having to check the last block replaced causing more over head.

With the 256 KB cache size configuration we would notice some of the best hit rates where all of the simulations had a 90% and higher hit rate, the draw back with this cache size is we noticed that for all the various block sizes and associativity we would have very high amounts of unused cache space. This would result in unnecessary amounts of waste in cost for almost every variation with the cache size. Once we got to 8-way and 16-way associativity we would see some of the best results with the cache size with very minimal miss rates, less than 2%. The biggest draw back with the 256 KB cache size is the cost per unit as we jump to $13.25 and higher for each variation compared to the most expensive cost for a 64 KB cache size is $5.20. As appealing as the higher hit rate and lower CPI may be, the wasted cache space and unused cache just does not make this a viable option.

With the 64 KB cache size configuration we saw some of the best overall results looking at key components such as cost per unit, hit rate, and wasted cache space. Between the three different block sizes, the best performance would come from having 16-way associativity having the best hit rate and CPI. As we do increase block size, we do see a jump in hit rate as well going from 89% for 4 byte blocks, 96% for 16 byte blocks, and 98% in 64 byte blocks. Though with the higher hit rate we do go up little bit little in CPI. The smaller associativity sizes would yield higher cost per unit as well as higher miss rates making them not very viable options compared to 16-way associativity.

# Test Results



# SuggEstED aCTIONS

After all of the testing, simulations, and data analysis the most viable option to select for the cache configuration would be 64 KB cache size, 16 byte block size, 16-way associativity, and the random replacement method. This configuration shows the best results when you weigh in key component factors as cost per unit, CPI, hit rate, and unused cache space. This configuration has a 3.0 CPI, cost $3.30 per unit, a hit rate of 95%, and does not waste and cache space as it uses all of the available cache space. Compared to the 4 byte block size configuration, the extra $1.50 per unit does not outweigh the .21 higher CPI jump, on top of the higher cost you will have a significant drop in hit rate by almost 7% as well. The 64 byte configuration does have the best hit rate, 98%, out the 2 other options stated before and the lowest cost at $3.30 but does have quite a higher jump in CPI up to 3.6 versus the 3.0 of the 16 byte configuration. The extra $0.40 per unit will over all improve performance with the best combined hit rate, CPI, and un-wasted cache space.

APPENDIX A: REFERENCES

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Description** | **Location** |
| FinalTrace.csv | Spreadsheet of the 240 simulations ran with each different configuration. | Located in zip folder that contains this report |

APPENDIX B: KEY TERMS

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Associativity | Provides N set of blocks per index and row in the cache |
| Replacement policy | The policy that will dictate on how to replace a cache block in a row on a conflict miss. Round robin – cycle through each block numerically, i.e. 1, 2, 3, 4, 1, 2,…. Random – select a block in that row at random. Least Recently Used – replaces that first used block in that row. |
| Compulsory Miss | When a block of main memory is trying to occupy a fresh empty block of cache. |
| Conflict Miss | When every block in a matching row of cache memory is filled but does not match the tag of the current access to memory, so a block of cache needs to be replaced. |
| CPI | Cycles per instruction, the average number of clock cycles per instruction |

APPENDIX C: FORMULAS FOR CALULATIONS

The following table provides formulas for calculations relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Formula** |
| Total Number of Blocks | 2cache\_bits - offset |
| Cache Bits | Log2(cache\_size \* 1000) |
| Offset | Log2(block\_size) |
| Index Bits | cache\_bits – (offset + Log2(associativity) |
| Tag Bits | 32 – index\_bits - offset |
| Total Number of Rows | 2index\_bits |
| Overhead Size | 2index\_bits \* (tag\_bits + 1) / 8 \* associativity |
| Total Implementation Size | 2cache\_bits + overhead\_size |
| Cost | (Total\_implementation\_size \* 0.0009765625) \* .05 |
| Unused Cache Blocks | Block\_count – compulsory\_misses |
| Unused Cache Space | (Unused\_cache\_blocks \* (((block\_size \* 8) + tag\_bits + 1) / 8)) / 1024 |
| Waste | Unused\_cache\_space \* .05 |
| Total Cache Size | Total\_implementation\_size \* 0.0009765625 |