CS224 Computer Organization

Preliminary Report

Lab 06

Section 2

Burak Ozturk

21901841

# Part 1.1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Cache**  **Size KB** | **N way**  **cache** | **Word**  **Size**  **in bits** | **Block size**  **(no. of words)** | **No. of**  **Sets** | **Tag Size**  **in bits** | **Index Size**  **(Set No.) in bits** | **Word Block**  **Offset**  **Size in bits1** | **Byte**  **Offset**  **Size in bits2** | **Block**  **Replacement**  **Policy Needed (Yes/No)** |
| 1 | 128 | 1 | 32 | 4 | 213 | 15 | 13 | 2 | 2 | No |
| 2 | 128 | 2 | 32 | 4 | 212 | 16 | 12 | 2 | 2 | Yes |
| 3 | 128 | 4 | 32 | 8 | 210 | 17 | 10 | 3 | 2 | Yes |
| 4 | 128 | Full | 32 | 8 | 20 | 27 | 0 | 3 | 2 | Yes |
| 9 | 256 | 1 | 16 | 4 | 215 | 19 | 15 | 2 | 1 | No |
| 10 | 256 | 2 | 16 | 4 | 214 | 15 | 14 | 2 | 1 | Yes |
| 11 | 256 | 4 | 16 | 16 | 211 | 16 | 11 | 4 | 1 | Yes |
| 12 | 256 | Full | 16 | 16 | 20 | 27 | 0 | 4 | 1 | Yes |

# Part 1.2

|  |  |  |
| --- | --- | --- |
| **Memory Address Accessed (hex)** | **Set No.** | **Hit (yes/no)** |
| 00 00 00 24 | 00 | no |
| 00 00 00 42 | 00 | no |
| 00 00 00 68 | 01 | no |
| 00 00 00 04 | 00 | no |
| 00 00 00 0C | 01 | no |
| 00 00 00 4C | 01 | no |

# Part 1.3

|  |  |  |
| --- | --- | --- |
| **Memory Address Accessed (hex)** | **Set No.** | **Hit (yes/no)** |
| 00 00 00 2C | 01 | no |
| 00 00 00 48 | 01 | no |
| 00 00 00 44 | 00 | no |
| 00 00 00 0C | 01 | no |
| 00 00 00 04 | 00 | no |
| 00 00 00 0C | 01 | yes |

# Part 1.4

## L1

Hit rate: 80%, Miss rate: 20%

Access time: 1 cycle

## L2

Effective hit rate: (L1 Effective Miss Rate) \* (L2 Hit Rate) = 20% \* 95% = 19%

Effective miss rate: (L1 Effective Miss Rate) \* (L2 Miss Rate) = 20% \* 5% = 1%

Access time: 2 cycles

Effective Access time: L1 access time + 2 = 3 cycles

## L3

Effective hit rate: (L2 Effective Miss Rate) = %1

Effective miss rate: 0

Access time: 20 cycles

Effective Access time: L2 effective access time + 20 = 23 cycles

AMAT = (20 \* 1 + 19 \* 3 + 1 \* 23) / 100 = 1.6 cycles

Runtime = #instructions \* #AMAT / Clock Rate (Hz) = (1.6\*1012)/(4\*109) = 400 secs