
# İhsan Doğramacı Bilkent University

## Computer Science

## Computer Organization

## CS 224

## Lab 6

## Section 1

## Ömer Oktay Gültekin

## 21901413

## 25/04/2022

**1)** 2 GB => Memory Address 31 bit (All below should sum up to 31)

| No. | Cache Size KB | N way cache | Word Size (no. of bits) | Block size (no. of words) | No. of Sets | Tag Size in bits | Index Size (Set No.) in bits | Block Offset Size in bits[1] | Byte Offset Size in bits[2] | Block Replacement Policy Needed (Yes/No) |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 64 | 1 | 32 | 4 | 4096 | 15 | 12 | 2 | 2 | No |
| 2 | 64 | 2 | 32 | 4 | 2048 | 16 | 11 | 2 | 2 | Yes |
| 3 | 64 | 4 | 32 | 8 | 512 | 17 | 9 | 3 | 2 | Yes |
| 4 | 64 | Full | 32 | 8 | 1 | 26 | 0 | 3 | 2 | Yes |
| 9 | 128 | 1 | 16 | 4 | 16384 | 14 | 14 | 2 | 1 | No |
| 10 | 128 | 2 | 16 | 4 | 8192 | 15 | 13 | 2 | 1 | Yes |
| 11 | 128 | 4 | 16 | 16 | 1024 | 16 | 10 | 4 | 1 | Yes |
| 12 | 128 | Full | 16 | 16 | 1 | 26 | 0 | 4 | 1 | Yes |

[1] **Block Offset Size in bits:** $\text{Log}_2$(No. of words in a block)
[2] **Byte Offset Size in bits:** $\text{Log}_2$(No. of bytes in a word)

**2.** Consider the following MIPS code segment. (Remember MIPS memory size is 4 GB.) Cache capacity is 16 words, Block size: 4 words, N= 2.

```
        addi    $t0, $0, 5
loop:   beq     $t0, $0, done
        lw      $t1, 0x24($0)
        lw      $t2, 0xAC($0)
        lw      $t3, 0xC8($0)
        addi    $t0, $t0, -1
        j       loop
done:
```

**a.**

| Instruction | Iteration No. | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| lw $t1, 0x24($0) | Compulsory | Conflict | Conflict | Conflict | Conflict |
| lw $t2, 0xAC($0) | Compulsory | Conflict | Conflict | Conflict | Conflict |
| lw $t3, 0xC8($0) | Compulsory | Conflict | Conflict | Conflict | Conflict |

**b.** What is the total cache memory size in number of bits? Include the V bit your calculations. Show the details of your calculation.

Tag = 32(MIPS Mem Addr.) – 1(S bit) – 2(Block Offset) – 2(Byte Offset) = 27 Bit

Total Cache Memory Size = S (# of sets) * N (Associavity degree) * (V bit + Tag + Block Size * Word Size) = 2 * 2 * (1 + 27 + 4 * 32) = **624 bits**

**c.** State the number of AND and OR gates, EQUALITY COMPARATORs and MULTIPLEXERs needed to implement the cache memory. No drawing is needed.

Number of And Gates: 2
Number of Or Gates: 1
Number of EQUALITY COMPARATOR: 2
Number of MULTIPLEXER: 3 (2 4:1 MUX for choosing correct word from blocks + 1 2:1 MUX for choosing correct word from the outputs of 4:1 MUX)

**3**. Consider the above MIPS code segment. Block size is 1 word. There is only 1 set. Cache memory size is 2 blocks. The block replacement policy is LRU.

**a.**

| Instruction | Iteration No. | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| lw $t1, 0x24($0) | Compulsory | Capacity | Capacity | Capacity | Capacity |
| lw $t2, 0xAC($0) | Compulsory | Capacity | Capacity | Capacity | Capacity |
| lw $t3, 0xC8($0) | Compulsory | Capacity | Capacity | Capacity | Capacity |

**b.** How many bits are needed for the implementation of LRU policy? What is the total cache memory size in number of bits? Include the V bit and the bit(s) used for LRU in your calculations.
1 bit is needed to implement the LRU policy since there is 2 different way and 1 bit is enough for representing 2 different way.
Tag = 32(MIPS Mem Addr.) – 2(Byte Offset) = 30 Bit

Total Cache Memory Size = U (Overhead) + S (# of sets) * N (Associavity degree) * (V bit + Tag + Block Size * Word Size)
= 1 + 1 * 2 * 1 * (1 + 30 + 1 * 32) = **127 bits**

**c.** State the number of AND and OR gates, EQUALITY COMPARATORs and MULTIPLEXERs needed to implement the cache memory. No drawing is needed.
Number of And Gates: 2
Number of Or Gates: 1
Number of EQUALITY COMPARATOR: 2
Number of MULTIPLEXER: 1 (1 2:1 MUX for choosing correct word coming from ways)

**4**. a)Consider a three level memory: L1 and L2 are for cache memory and the third level is for the main memory. Access time for L1 is 2 clock cycle, the access time for L2 is 4 clock cycles and main memory access time is 20 clock cycles. The miss rate for L1 is 10% and the miss rate for L2 is 5%. What is the effective clock cycle for memory access (AMAT in number of clock cycles)?

General formula AMAT = Time for a hit + Miss Rate (MR) * Miss Penalty
We have 2 caches, therefore, need to calculate above formula for each of them
$MR_{L1}$ = Miss Rate for L1 = 10%, $MR_{L2}$ = 5%, $t_{L1}$ = 2 Clock Cycle, $t_{L2}$ = 4 Clock Cycle, $t_{MM}$ = 20 Clock Cycle

$AMAT = t_{L1} + MR_{L1} * (t_{L2} + MR_{L2} * (t_{MM})) = 2 + 0.1 * (4 + 0.05 * 20) = 2.5$ = Effective Clock Cycle for Memory Access

b) With 2 GHz clock rate how much time is needed for a program with $10^{10}$ instructions to execute?

$T_{exec}$ = (No. of Inst.) * (CPI) * $T_c$

$T_c$ = Clock Period = 1 / clock rate = 1 / ($2 * 10^9$) = $5 * 10^{-10}$ sec

CPI = Cycles Per Instruction = 2.5 (% of Load/Store Instructions not given. I assume all instructions are Load/Store)

$T_{exec} = 10^{10} * 2.5 * 5 * 10^{-10} = 12.5$ sec