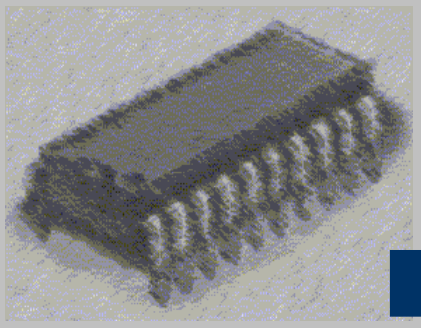
Exercises

Lesson 4: The memory System

Computer Structure and Organization



Graduate in Computer Sciences
Graduate in Computer Engineering

1. Let's be a 32 bits computer with the following memory system.
    - Main memory with 32 ns. access time.
    - Split caches (instruction and data) Block size: 16 bytes. 4 ns. access time. Cache capacity 64 Kbytes each one.
    - Both cache memories use direct mapping and write through.

    Next code is executed in this computer:

```
WHILE (i ≤ nprod) DO
BEGIN
        Total := Total + cost[i] * unit[i];
        i:= i + 1;
END;
```

If data cache is empty:

a.- Calculate main memory and cache addresses correspondence.

b.- Trace the first four memory accesses of above program

- ***i*** and ***Total*** variables are stored in microprocessor registers, and set to 0.
- ***cost*** and ***unit*** are stored in memory from 3001Ah and 9001Ah memory position respectively.

c.- Which ones of the previous memory accesses are failures?

d.- Calculate the hit ratio for above trace program.

---

2. Solve previous exercise but suppose both caches use set-associative mapping of two blocks per set and replacement policiy is FIFO.

---

3. Let's be following memory system:
   - Interleaved simple main memory of 16 modules with 100 ns. access time
   - Unified cache memory:
     - Capacity: 128 Kwords
     - Access time: 20 ns.
     - Set-associative mapping of 2 blocks. Block size 16 words.
     - Replacement policy: LIFO
     - Write policy: write through without allocation

Next program is executed on this computer:

Instruction Main Memory Address	Datum Main Memory Address	Pseudo code	
011110 h		Call AskData.	Main Program
011112 h		Call CodeData	
031110 h		Display message asking for data	AskData Procedure
031112 h	001119 h	Stored on ReadDatum from keyboard	
031114 h		Return to main program	
041110 h	001119 h	ReadDatum = ReadDAtum XOR 70h	CodeData Procedure
041112 h	001119 h	ReadDAtum = ReadDAtum XOR 8000h	
041114 h		Return to main program	

Following operations need read and write of ReadDatum variable:

- ReadDatum := ReadDatum XOR 70h;
- ReadDatum:= ReadDatum XOR 8000h;

IF cache memory is empty and ***ReadDatum*** variable is stored 001119 h main memory address.

a.- Calculate main memory and cache addresses correspondence.

b.- Trace the first four memory accesses of above program. Which ones of the previous memory accesses are failures?

c.- Calculate hit ratio and execution time referred to memory accesses only.

---

4. Solve previous exercise but suposse that write policy is write through with allocation.

---

5. Let's be a computer with following memory system characteristics
   - Interleaved simple main memory with 16 modules and an access time 20 ns.
   - Memoria caché unificada (común para instrucciones y para datos), con las siguientes características:
   - Unified cache memory:
     - Capacity: 128 Kwords
     - Access time: 20 ns.
     - Set-associative mapping of 2 blocks. Block size 16 words.
     - Replacement policy: FIFO
     - Write policy: copy back without allocation

Next program is executed on this computer:

Instruction Main Memory Address	Datum Main Memory Address	Pseudo code	
0001 0004 h		Call AskData.	Main Program
0001 0008 h		Call CodeData	
0008 0002 h		Display message asking for data	AskData Procedure
0008 0006 h	0000 0009 h	Stored on ReadDatum from keyboard	
0008000A h		Return to main program	
0006 0000 h	0000 0009 h	ReadDatum = ReadDAtum XOR 70h	CodeData Procedure
0006 0004 h	0000 0009 h	ReadDAtum = ReadDAtum XOR 8000h	
0006 0008 h		Return to main program	

Following operations need read and write of ReadDatum variable:

- ReadDatum := ReadDatum XOR 70h;
- ReadDatum:= ReadDatum XOR 8000h;

IF cache memory is empty and ***ReadDatum*** variable is stored 00000009h main memory address.

a.- Calculate main memory and cache addresses correspondence.

b.- Trace the first four memory accesses of above program. Which ones of the previous memory accesses are failures?

c.- Calculate hit ratio and execution time referred to memory accesses only.

---

6. Solve previous exercise but suppose a copy back with allocation write policy.

---

7. Let's be a computer with the following memory system:

- Main memory:
  - Capacity: 4 Mbytes
  - Access time: 64 ns
  - Interleaved simple with 16 modules
- Cache memory:
  - Unified cache
  - Capacity: 16 Kbytes
  - Block size: 16 bytes
  - Access time: 9 ns
  - Set-associative mapping of 4 blocks per set
  - Write policy: write through without allocation.
  - Replacement policy: FIFO

Next program is execute in this computer:

Memory address	Instruction	
5010h	Call Procedure 1	Main program
5014h	Call Procedure 2	
5018h	Call Procedure 3	
6014h	Read a key from keyboard	Procedure 1
6018h	Store it on ReadDatum	
601Ch	Return	
7010h	ReadDAtum = ReadDAtum -30	Procedure 2
7014h	Return	
8018h	Addition = Addition + ReadDatum	Procedure 3
801Ch	Return	

- Following operations need read and write of **ReadDatum** and **Addition** variabls:
- ReadDatum := ReadDatum - 30h;
- Addition := Addition + ReadDAtum;

Addition and ***ReadDatum*** variables are stored in 2001Ch and 30014h memory addresses.

a.- Calculate main memory and cache addresses correspondence.

b.- Trace the first four memory accesses of above program. Which ones of the previous memory accesses are failures?

c.- Calculate hit ratio and execution time referred to memory accesses only.

---

8. Solve previous exercise with a copy back with allocation write policy.

---

9. Let's be a computer with the following memory system:

- Main memory:
  - Capacity: 4 GB.
  - 16 modules simple interleaved main memory.
  - Access time: 32 ns.
- Cache memory:
  - Split cache memory
  - Cache capacities: 256 Kbytes each one
  - Access time: 4 ns.
  - 16 bytes per block
  - **Set-associative mapping**: 4 blocks per set
  - **Write policy**: write through without allocation.
  - **Replacement policy**: FIFO

Next code will be executed on this computer:

```
i := 1;
  WHILE (i < 3) DO
  BEGIN
        A  := Vector1(i);
        B:= Vector2(i);
        IF A > B
        THEN BEGIN
                Mayor(i) := A;
                Menor(i) := B;
        END
        ELSE BEGIN
                Mayor(i) := B;
                Menor(i) := A;
        END;
        Suma(i) := Suma(i) + Mayor(i);
        i:= i + 1;
  END;
```

Instruction *Suma(i) := Suma(i)* + *Mayor(i)* needs three memory accesses: two reading ones *suma(i)* y *mayor(i)* and one writing access *suma(i)*

i variable is set to 1 and it's allocated in a microprocessor register

Variables are in following main memory addresses:

Mayor	3C89 2365	Vector1	0039 2364
Menor	3D89 236A	Vector2	0029 2368
Suma	0019 2367		

If cache memory is empty:

a.- Calculate main memory and cache addresses correspondence.

b.- Trace the first four memory accesses of above program. Which ones of the previous memory accesses are failures?

c.- Calculate hit ratio and execution time referred to memory accesses only.