

ESTRUCTURA i TECNOLOGIA de COMPUTADORS

Transparències del Tema 1 Introducció i Conceptes Bàsics

J. Freixenet, X. Cufí, J. Martí,
M. Fàbregas i J. Ferrer
Departament d'Electrònica,
Informàtica i Automàtica

1

ETC - Esquema del Tema 1

Presentació del curs

Conceptes bàsics sobre computadors

L'ordinador com a màquina programable

Hardware, Software (programes i instruccions)

Estructura bàsica d'un computador

CPU's, Memòries, dispositius d'E/S, i el paper del *clock*

Execució d'una instrucció: primera aproximació

Representació de la informació

2

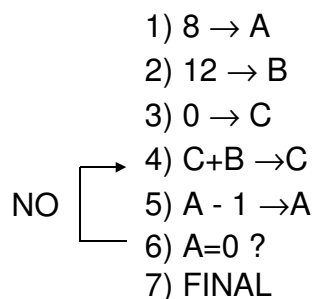
ETC - Conceptes bàsics sobre computadors

COMPUTADOR DIGITAL: Màquina que pot resoldre problemes executant instruccions. Si es canvien les instruccions es pot adaptar la màquina a solucionar problemes diferents (→ Programar!).

PROGRAMA: Seqüència d'instruccions que pot executar una determinada màquina i que descriu com s'ha d'executar una determinada tasca.

Exemples: La rentadora de rentar roba, la realització d'una truita, multiplicar fent sumes.

Multiplicació 8x12 sumant:



3

ETC - Conceptes bàsics sobre computadors

INSTRUCCIÓ: Una operació bàsica, que pot ser executada per un computador.

HARDWARE/SOFTWARE: (*Hard* i *Soft* a partir d'ara)

HARD: Part material, física, tangible que s'ocupa de l'execució dels programes (→ Són els circuits electrònics, els cables, els xips, ...).

SOFT: Conjunt d'algorismes, programes, instruccions que expliquen com realitzar una determinada tasca.

Implementar amb Hard o Soft? A voltes es pot dissenyar *Hard* específic per resoldre determinats problemes de forma més ràpida. També es pot solucionar el mateix problema per *Soft*, utilitzant una màquina de propòsit general.

4

ETC - Conceptes bàsics sobre computadors

LLENGUATGE de PROGRAMACIÓ: Conjunt de símbols (paraules) + regles sintàctiques que permeten expressar de forma clara i concisa la forma de resoldre un problema. Un llenguatge de programació permet escriure un programa.

LLENGUATGE MÀQUINA: El conjunt d'instruccions que és capaç d'executar un computador s'anomena *Llenguatge Màquina*. La programació en *Llenguatge Màquina* és complexa i poc pràctica (→ feixuc per les persones).

LLENGUATGE D'ALT NIVELL: Normalment una instrucció de *Llenguatge d'Alt Nivell* equival a varies instruccions de *Llenguatge Màquina*.

Facilita al programador la creació de programes ja que els LAN són "propers" a la forma de pensar humana. Aquests programes hauran de ser "traduïts" a *Llenguatge Màquina*, que és l'únic llenguatge que entén el computador.

CONVERSIÓ D'UN PROGRAMA D'ALT NIVELL A LLENGUATGE MAQUINA

Dues alternatives: Traduir (compilar) o interpretar.

Traduir: Convertir tot el programa, corregint errors lèxics i sintàctics. Es genera un nou programa: l'escrit en *Llenguatge Màquina* (en MS-DOS → .EXE).

Interpretar: Es tradueix a codi màquina instrucció rera instrucció durant l'execució. 5

ETC - Conceptes bàsics sobre computadors

EXEMPLE DE LLENGUATGES DE PROGRAMACIÓ:

High level lang.	Assembler lang.		Machine lang.			
	label	remarks	@	cod. op.	@ source	@ desti
begin						
c := 0;	begin :	MOV zero,c ; c:= 0	0	2	105	102
i := 0;		MOV zero,i ; i:= 0	1	2	105	103
while i < b do	while:	CMP i,b ; while i < b	2	1	103	101
begin		BEQ end	3	3	x	8
c := c + a;		ADD a,c ; c:= c + a	4	0	100	102
i := i + 1		ADD one,i ; i:= i + 1	5	0	104	103
end		CMP X,X	6	1	105	105
end.		BEQ while	7	3	x	2
	end:		8			

ETC - Estructura bàsica d'un computador

ESTRUCTURA (ARQUITECTURA) D'UN COMPUTADOR:

Estructura Von Neumann = CPU + Memòria + Dispositius d'Entrada/Sortida

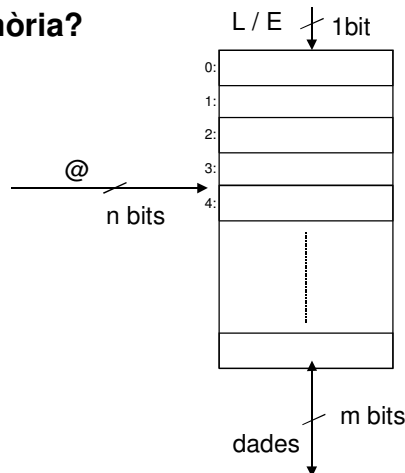
MEMÒRIA: Magatzem d'informació.

Com s'organitza? En forma de seqüència de cel·les, on a cada cel·la s'hi guarda un número i cada cel·la té una adreça.

Quines operacions s'hi fan? Llegir i escriure. En una memòria, en un instant només, es pot realitzar una operació sobre una sola cel·la.

Com són les cel·les?

Com es caracteritza una memòria?



7

ETC - Estructura bàsica d'un computador

CAPACITAT DE LES MEMÒRIES:

Capacitat d'emmagatzemar = 2^n cel·les x m bits

TIPUS DE MEMÒRIES:

Memòries RAM (Random Acces Memory): Memòries de lectura i escriptura.

Són volàtils: Només conserven el seu contingut si estan alimentades.

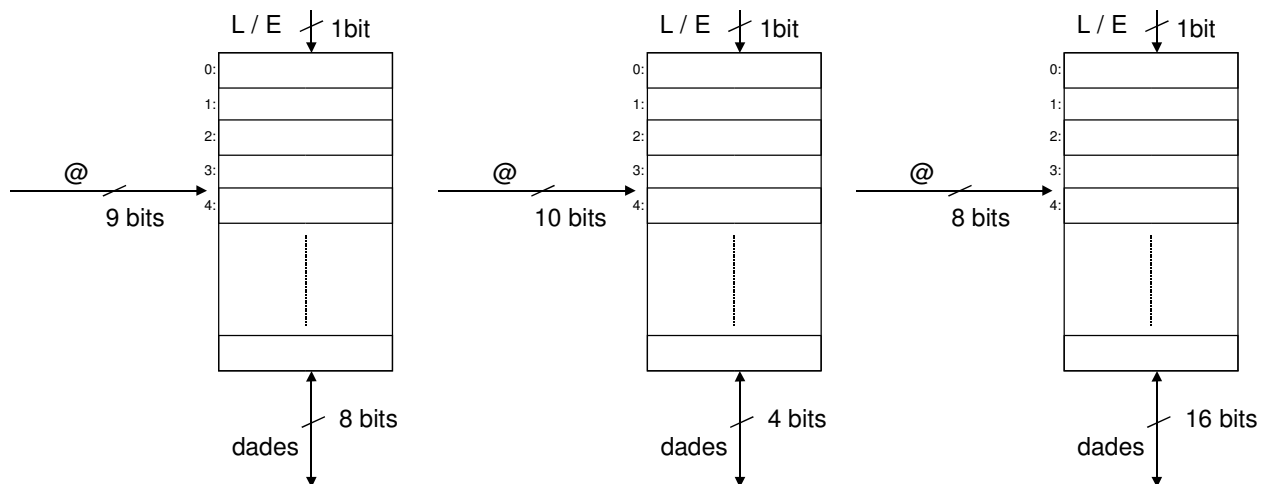
Què vol dir que una memòria sigui d'accés aleatori? Entre dues operacions consecutives de L/E sobre una memòria es pot accedir lliurement a qualsevol posició de memòria (el concepte contrari és l'accés seqüencial).

Memòries ROM (Read Only Memory): Memòries de només lectura. També són d'accés aleatori, i a més són *NO volàtils*; conserven el seu contingut sempre.

8

ETC - Estructura bàsica d'un computador

EXEMPLES DE MEMÒRIES: Possibles configuracions d'una memòria de 4096 Bits



9

ETC - Estructura bàsica d'un computador

CPU (Central Process Unit): És el processador del computador, el *xip* que s'encarrega pròpiament d'executar les instruccions de LLM (*Llenguatge Màquina*).

Es descomposa en *Unitat de Control (UC)* i *Unitat de Procés (UP)*.

La UC: S'encarrega d'interpretar les instruccions. Què vol dir interpretar?

- 1) Identificar quina és la instrucció a executar.
- 2) Determinar què cal fer per executar-la. És a dir, especificar les passes a seguir per executar-la.

La UP: És el conjunt format per la **ALU** (*Arithmetic Logic Unit*), els **RI** (*Registres Interns*) i la lògica de les interconnexions.

ALU: Hard (*Sistema Combinacionals*) que s'encarreguen de fer operacions aritmètiques (+, -, *, /, ...) i lògiques (*and, or, not, ...*).

Registres interns: Dispositiu que serveix per emmagatzemar un número. Sobre un registre es poden fer dues operacions: llegir i escriure (similar a les memòries, però tecnològicament i de naturalesa diferents).

Cada CPU té varis RI, tots els que es requereixin per tal d'emmagatzemar temporalment la informació necessària per executar qualsevol instrucció.

10

ETC - Estructura bàsica d'un computador

CONCEPTES SOBRE CPUS

Registres interns de propòsit general: el programador “els veu”, els pot utilitzar.

Registres interns de propòsit específic: són registres que serveixen exclusivament per emmagatzemar una informació específica.

Exemples:

IR (*Instruction Register*): Emmagatzema la instrucció a executar o que s'està executant.

PC (*Program Counter*): Punter de programa. Indica quina és la següent instrucció a executar.

CPU's RISC (*Reduced Instruction Set Computer*): Tendència actual i de futur. Poques instruccions i molt ràpides d'executar.

CPU's CISC (*Complex Instruction Set Computer*): Moltes instruccions diferents que, a més a més, realitzen operacions molt complexes.

11

ETC - Estructura bàsica d'un computador

DISPOSITIUS d'E/S (perifèrics): Bloc que permet comunicació amb l'exterior.

Un computador és un sistema obert: han de poder entrar i sortir dades (informació). Les operacions d'entrada corresponen a llegir, mentre que les de sortida corresponen a escriure.

Perifèrics d'entrada: teclat, ratolí, *scanner*, *joystick*, ...

Perifèrics de sortida: pantalles, impressora, altaveus, ...

Perifèrics d'entrada i sortida: disquets, disc durs, mòdems, pantalles tàctils, ...

BUSOS: Camins que permeten la comunicació entre els diferents mòduls del computador (*CPU*, *Memòria*, *E/S*). És un medi de comunicació compartit per varis dispositius. Tots tenen accés al bus, però lògicament cal arbitrar l'accés per evitar col·lisions.

Els busos s'organitzen segons la informació que “transporten”:

Bus adreces (*unidireccional*): Indica la posició de memòria o selecciona un perifèric per fer una operació.

Bus de dades (*bidireccional*): Línies que transporten les dades que es llegeixen o escriuen en una determinada adreça.

Bus de control: Control de les operacions. Per exemple, especificar si cal llegir o escriure, a memòria o a perifèric, ...

12

ETC - Estructura bàsica d'un computador

EXECUCIÓ D'UNA INSTRUCCIÓ: PRIMERA APROXIMACIÓ

1. Imaginem un PC amb un sistema operatiu de finestres, ...
2. Imaginem un usuari que fa un doble clic sobre una icona de l'escriptori anomenada "tetris.exe".
3. Què passa? D'entrada aquest fitxer que "viu" al disc dur i és pròpiament un programa escrit en LLM, es carrega a la Memòria (→ operacions de lectura al disc i escriptura a memòria).
4. Un cop es disposa de *tot* el programa carregat a memòria, la CPU llegeix la 1era instrucció del programa i la càrrega al registre d'instrucció de dins la UP.
5. La UC interpreta la instrucció, la identifica i es determina què cal fer per executar-la. Imaginem que es tracta d'una instrucció que preten sumar dos números. Per tant, la UC planifica que cal llegir dos números (1er un i després l'altre), després sumar-los, i finalment emmagatzemar el resultat.
6. Es porta a terme el pla de la UC, de forma que s'executa la instrucció que es troba dins el registre d'instrucció.
7. S'ha acabat d'executar una instrucció!
8. Anem per la següent, i d'aquesta manera anar fent seqüencialment.

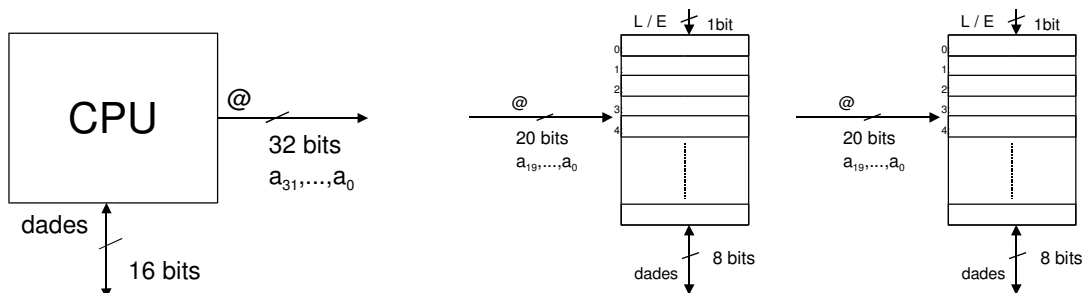
13

ETC - Estructura bàsica d'un computador

RECORDATORI MEMÒRIES (I)

Imaginem un computador amb una CPU capaç d'adreçar fins a 8Gb, però amb una memòria RAM real de 2Mb.

Possibles configuracions (I):



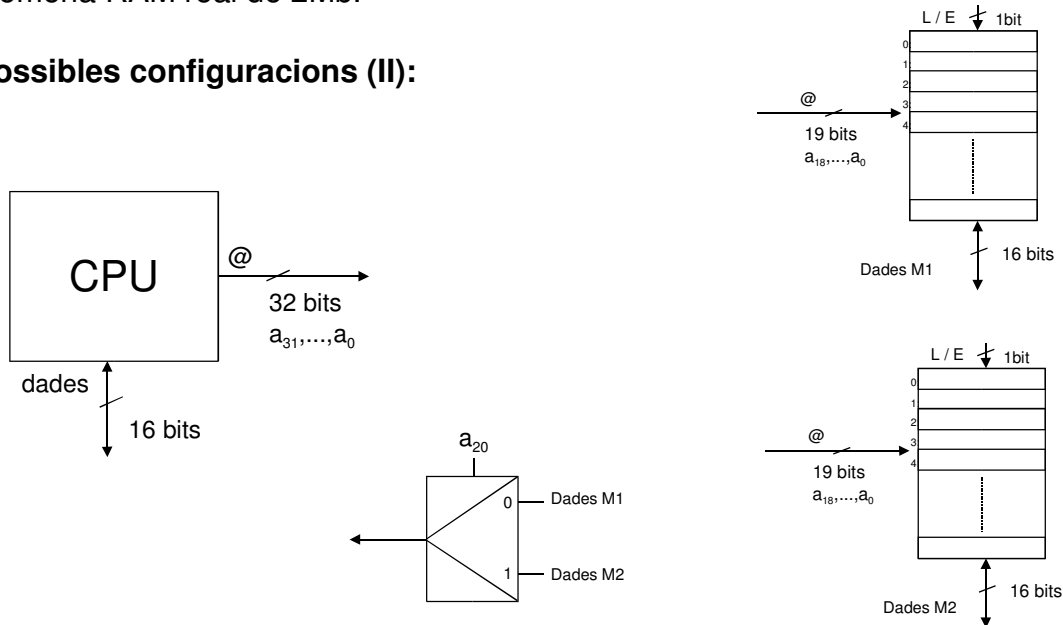
14

ETC - Estructura bàsica d'un computador

RECORDATORI MEMÒRIES (II)

Imaginem un computador amb una CPU capaç d'adreçar fins a 8Gb, però amb una memòria RAM real de 2Mb.

Possibles configuracions (II):



15

ETC - Representació de la informació

Hem definit computador com a màquina que pot solventar problemes, llegint, escrivint dades, informació. *Però, com és aquesta informació?*

Sistema numèric d'interés: **El Codi Binari**

- 2 símbols { 0, 1 }
- Cada xifra, cada dígit s'anomena bit (**b**inary **d**igit)
- **IMPORTANT!** Circuits digitals electrònics poden treballar fàcilment només en dos estats: obert - tancat, passa llum - no en passa, passa corrent - no en passa, ...

Presentar números en binari, d'un bit, de dos bits,...

- Números de 8 bits \rightarrow 1 byte
- 1 kbyte = 1024bytes = 2^{10} bytes
- 1 kbyte \cong 1000bytes
- 1 Mbyte (Megabyte) = 1024 kb
- 1 Mb \cong 1000kb
- 1 Gb (Gigabyte) = 1024 Mb

Altres sistemes numèrics d'interés: **El codi hexadecimal** = 16 símbols diferents

16

Recordem que:

BASE = Número de símbols de l'alfabet.

SÍMBOL = Cada element de l'alfabet.

DÍGIT = Símbol situat en una determinada posició dins un número.

La posició = el pes. Així parlarem del dígit de més pes (més a l'esquerra) i el de menys pes (més a la dreta).

Valor d'un número a partir de la posició que ocupen els seus dígit:

$$N = A_n B^n + A_{n-1} B^{n-1} + \dots + A_1 B^1 + A_0 B^0 + A_{-1} B^{-1} + \dots + A_{-p} B^{-p}$$

On **n + 1** és el número de xifres senceres, **p** és el número de xifres fraccionàries i **B** és la base de l'alfabet.

Exemples:

N = 387,45₁₀

$$N = 3 \cdot 10^2 + 8 \cdot 10^1 + 7 \cdot 10^0 + 4 \cdot 10^{-1} + 5 \cdot 10^{-2} = \mathbf{387,45}_{10}$$

N = 1101,01₂

$$N = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = \mathbf{13,25}_{10}$$

17

CANVIS DE BASE

- N en base B → M en base 10
- N en base 10 → M en base B

Utilitzar el polinomi equivalent.

Utilitzar mètode divisions i productes successius.

Exemples: 153₁₀ → Base 8 ?

$$\begin{array}{r} 153 \overline{)8} \\ 73 \ 19 \overline{)8} \\ \underline{1} \quad \underline{3} \quad \underline{2} \end{array} \rightarrow 231_8$$

0,6875₁₀ → Base 2 ?

$$\begin{array}{l} 0,6875 \times 2 = 1,3750 \rightarrow \textcircled{1} \\ 0,3750 \times 2 = 0,7500 \rightarrow \textcircled{0} \\ 0,7500 \times 2 = 1,5000 \rightarrow \textcircled{1} \\ 0,5000 \times 2 = 1,0000 \rightarrow \textcircled{1} \end{array} \rightarrow 0,1011_2$$

La darrera divisió és exacte: lògicament això no passa sempre. Llavors cal decidir quants decimals cal agafar.

18

ETC - Representació de la informació

REPRESENTACIÓ DE NÚMEROS SENCERS (I)

SIGNE I MAGNITUD:

S'inclou un bit que fa de signe (0 → Positiu, 1 → Negatiu)

Rang de representació:

$[-(2^{n-1} - 1), 2^{n-1} - 1]$ essent n el número de bits

Problemes:

El número zero té dues representacions

Overflow

La suma no és homogènia

19

ETC - Representació de la informació

REPRESENTACIÓ DE NÚMEROS SENCERS (II)

COMPLEMENT A 1 (C'1):

Els números positius es representen com a **S+M**, mentre que els negatius es troben fent el C'1 del positiu.

Com es fa el C'1 d'un número? Substituint zeros per uns i uns per zeros.

Rang de representació:

$[-(2^{n-1} - 1), 2^{n-1} - 1]$ essent n el número de bits

Problemes:

El número Zero té dues representacions

Overflow

Característiques:

$C'1 (C'1 (a)) = a$

El *carry out* es suma al resultat

20

ETC - Representació de la informació

REPRESENTACIÓ DE NÚMEROS SENCERS (III)

COMPLEMENT A 2 (C'2):

Els números positius es representen com a **S+M**, mentre que els negatius es troben fent el C'2 del positiu.

Com es fa el C'2 d'un número? Amb dues passes:

1. Fent el C'1
2. Sumar 1

Rang de representació:

$[-(2^{n-1}), 2^{n-1} - 1]$ essent **n** el número de bits

Problemes:

Overflow

Característiques:

$C'2(C'2(a)) = a$

La suma és homogènia → és el mateix sumar números naturals, o números en C'2. El mateix sumador pot sumar números positius o números amb signe en C'2 (→ Compte amb l'*Overflow*)

21

ETC - Representació de la informació

REPRESENTACIÓ DE NÚMEROS FRACCIONARIS (I)

COMA FIXE:

Si es vol representar un número **N** amb **n** bits, es decideix donar **n_e** bits per a representar la part entera i **n_f** bits per a la fraccionaria, a més d'un bit per a representar el signe.

Llavors $n = n_e + n_f + 1$

Exemple: Representar en coma fixa el número **N = -4,327** amb **n_e = 5** i **n_f = 6**

El $4_{10} = 100_2$

El $0,327_{10} \rightarrow$

$0,327 \times 2 = 0,654$
$0,654 \times 2 = 1,308$
$0,308 \times 2 = 0,616$
$0,616 \times 2 = 1,232$
$0,232 \times 2 = 0,464$
$0,464 \times 2 = 0,928$

El número en coma fixa és:

1	00100	010100
---	-------	--------

Lògicament té un interval de representació, així com també cal tenir en compte que es comet un error

$$N = - (4 + 0,25 + 0,0625) = -4,3125$$

22

ETC - Representació de la informació

REPRESENTACIÓ DE NÚMEROS FRACCIONARIS (II)

COMA FLOTANT:

Si es vol representar un número N amb n bits, es decideix donar n_m bits per a representar la mantissa i n_e bits per a l'exponent, a més d'un bit per a representar el signe de la mantissa.

$$N = (\text{signe}) \text{ mantissa} \cdot B^{\text{exponent}} \text{ on } B = 2$$

Existeixen moltes representacions possibles → Normalitzar

- Mantissa amb part entera = 0
- Mantissa amb primera xifra a la dreta de la coma diferent de 0 (excepte pel número 0)
- Signe 0 → Positiu i 1 → Negatiu
- Exponent representat en C'2 (l'exponent ha de poder ser positiu o negatiu)

Exemple: Representar en coma flotant el número $N = -12,25$ amb $n_m = 9$ i $n_e = 4$

$$\text{El } 12,25_{10} = 1100,01_2 = 0,110001 \cdot 2^4$$

$$\text{L'exponent } 4_{10} = 100_2$$

El número en coma flotant és el

1	0100	110001000
---	------	-----------