

# Introducció als Ordinadors

---

Capítol 1

Característiques genèriques

Manel López

# Introducció als computadors

## DEFINICIONS PRÈVIES

- En un sentit genèric, un ORDINADOR és aquell aparell dissenyat per processar informació.
- Un procés es correspon a les fases o transformacions que suporta la informació per tal de resoldre un problema determinat.
- L'arquitectura d'un computador defineix el seu comportament funcional.

# Arquitectura i Organització

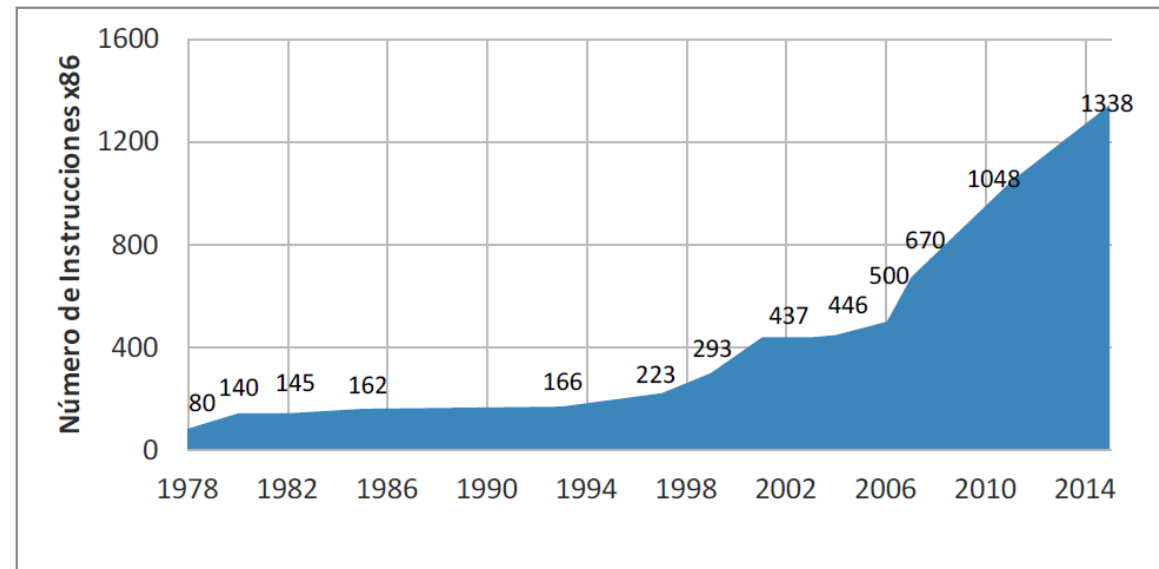
- **Arquitectura:** consisteix en aquells atributs que són visibles al programador
  - Conjunt d'instruccions
  - Nombre de bits usat per representació de dades
  - Mecanismes Entrada/Sortida
  - Tècniques d'adreça
  - **P.ex.: Hi ha instrucció de multiplicar?**
- **Organització:** consisteix en com s'han implementat, típicament amagats al programador
  - Senyals de control, interfícies, tecnologia de memòria
  - **P.Ex.: Hi ha unitat de multiplicació per HW o es fa per addició repetida (algoritme)?**

- Tota la família Intel x86 comparteixen la mateixa arquitectura bàsica
- La família System/370 comparteixen la mateixa arquitectura bàsica
- AMD i Intel comparteixen la mateixa arquitectura.
- Això proporciona **compatibilitat** de codi
  - Almenys cap enrere
  - Però la complexitat augmenta a cada generació. Potser seria més eficient començar nova arquitectura per a cada nova tecnologia e.g. RISC vs. CISC
- Organització difereix entre versions diferents

# Arquitectura Incremental

*Intel va apostar el seu futur a un processador d'altres prestacions. Aquest desenvolupament portaria anys. Per competir amb Zilog, Intel va desenvolupar un processador temporal anomenat 8086. La idea era que aquest processador durés poc temps al mercat i no tingués successors. Però la història no va ser així. El processador d'altres prestacions va arribar tard al mercat i era molt lent. D'aquesta manera el 8086 va seguir en el mercat i va evolucionar a un processador de 32 bits i eventualment a un de 64 bits. Els noms van anar canviant (8086, 80286, i386, i486, Pentium), però per qüestions de compatibilitat, el conjunt d'instruccions va quedar intacte...*

- Stephen P. Morse, arquitecte del 8086
- Intel presenta un ISA incremental, per tal d'obtenir compatibilitat de codi
- Moltes d'aquestes instruccions són obsoletes, però tot i això, es continuen mantenint

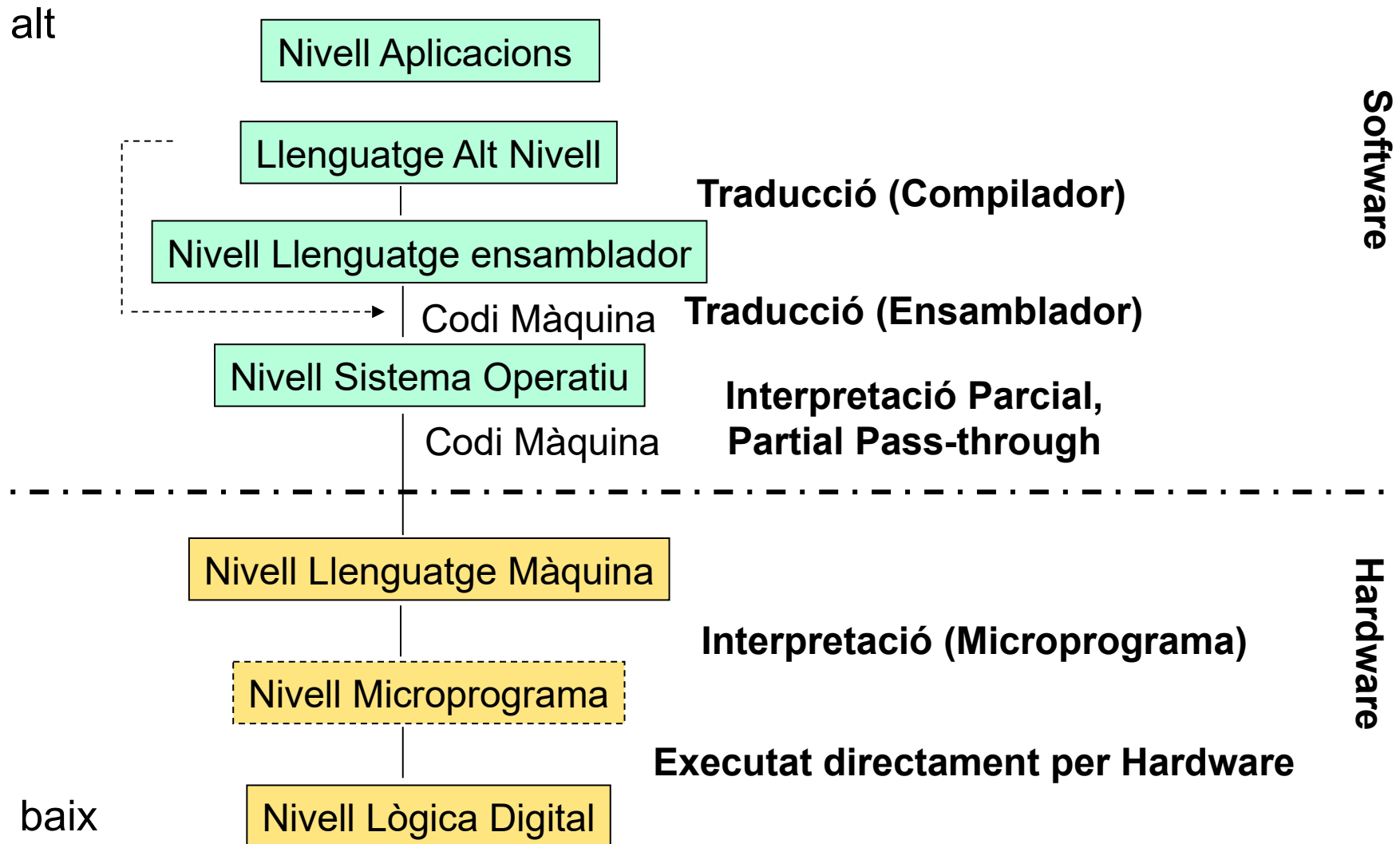


# Arquitectura Modular

- RISC – V un processador *open source* amb un ISA modular
- Incorpora un nucli fonamental anomenat RV32I que permet executar un stack de SW complet.
- La modularitat prové d'extensions opcionals estàndards que el HW pot incorporar en funció de les necessitats de cada aplicació.
- Cal notificar al compilador quines extensions existeixen en el HW.
- Exemple RV32IMFD incorpora:
  - Multiplicació
  - Punt flotant de 32b
  - Punt flotant de 64b

Base ISA	Instructions	Description
RV32I	47	32-bit address space and integer instructions
RV32E	47	Subset of RV32I, restricted to 16 registers
RV64I	59	64-bit address space and integer instructions, along with several 32-bit integer instructions
RV128I	71	128-bit address space and integer instructions, along with several 64- and 32-bit instructions
Extension	Instructions	Description
M	8	Integer multiply and divide
A	11	Atomic memory operations, load-reserve/store conditional
F	26	Single-precision (32 bit) floating point
D	26	Double-precision (64 bit) floating point; requires F extension
Q	26	Quad-precision (128 bit) floating point; requires F and D extensions
C	46	Compressed integer instructions; reduces size to 16 bits

# Computador com a màquina multi-nivell



# Nivells

## Nivell Lògica Digital

- Unitats Funcionals: (ALU; Registres,...).
- Nivell portes lògiques; Computacions primitives reduïdes a operacions Booleanes (AND, OR, NOT)
- Lògica combinacional. Registres LUTs,
- A aquest nivell no hi ha concepte de programa. Simplement seqüència d'operacions a ser processades

## Nivell Microprograma

- Instruccions de llenguatge màquina. Interpretades. Provoca una sèrie d'instruccions simples a ser executades en el nivell inferior Lògica-Digital
- *Microprograma* per cada instrucció, guardat permanentment a la memòria interna del Microprocessador



# Codificació en Binari

Representem la informació. Ex: Codificació ASCII

					0 1 0 1 1 0 0 1									
					0 0 1 1 1 1 0 0									
1	2	3	4	5	6	7	1	1	1	1	0	0	0	0
0	0	0	0				@	P	'	p	0	sp	NUL	DLE
1	0	0	0				A	Q	a	q	1	!	SOH	DC1
0	1	0	0				B	R	b	r	2	"	STX	DC2
1	1	0	0				C	S	c	s	3	#	ETX	DC3
0	0	1	0				D	T	d	t	4	\$	EOT	DC4
1	0	1	0				E	U	e	u	5	%	ENQ	NAK
0	1	1	0				F	V	f	v	6	&	ACK	SYN
1	1	1	0				G	W	g	w	7	'	BEL	ETB
0	0	0	1				H	X	x	x	8	(	BS	CAN
1	0	0	1				I	Y	y	y	9	)	HT	EM
0	1	0	1				J	Z	z	z	:	*	LF	SUB
1	1	0	1				K	[	k	{	;	+	VT	ESC
0	0	1	1				L	\	l		<	,	FF	FS
1	0	1	1				M	]	m	}	=	-	CR	GS
0	1	1	1				N	^	n	~	>	.	SO	RS
1	1	1	1				O	_	o	?	?	/	SI	US

Dec	Hx	Oct	Char	Dec	Hx	Oct	Htmi	Chr	Dec	Hx	Oct	Htmi	Chr	Dec	Hx	Oct	Htmi	Chr
0	0	000	NUL (null)	32	20	040	#32;	Space	64	40	100	#64;	R	96	60	140	#96;	.
1	1	001	SOH (start of heading)	33	21	041	#33;	"	65	41	101	#65;	A	97	61	141	#97;	a
2	2	002	STX (start of text)	34	22	042	#34;	"	66	42	102	#66;	B	98	62	142	#98;	b
3	3	003	ETX (end of text)	35	23	043	#35;	#	67	43	103	#67;	C	99	63	143	#99;	c
4	4	004	EOT (end of transmission)	36	24	044	#36;	&	68	44	104	#68;	D	100	64	144	#100;	d
5	5	005	ENQ (enquiry)	37	25	045	#37;	%	69	45	105	#69;	E	101	65	145	#101;	e
6	6	006	ACK (acknowledge)	38	26	046	#38;	@	70	46	106	#70;	F	102	66	146	#102;	f
7	7	007	BEL (bell)	39	27	047	#39;	'	71	47	107	#71;	G	103	67	147	#103;	g
8	8	010	BS (backspace)	40	28	050	#40;	"	72	48	110	#72;	H	104	68	150	#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	#41;	"	73	49	111	#73;	I	105	69	151	#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	#42;	"	74	4A	112	#74;	J	106	6A	152	#106;	j
11	B	013	VT (vertical tab)	43	2B	053	#43;	+	75	4B	113	#75;	K	107	6B	153	#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	#44;	,	76	4C	114	#76;	L	108	6C	154	#108;	l
13	D	015	CR (carriage return)	45	2D	055	#45;	-	77	4D	115	#77;	M	109	6D	155	#109;	m
14	E	016	SO (shift out)	46	2E	056	#46;	-	78	4E	116	#78;	N	110	6E	156	#110;	n
15	F	017	SI (shift in)	47	2F	057	#47;	/	79	4F	117	#79;	O	111	6F	157	#111;	o
16	10	020	DLE (data link escape)	48	30	060	#48;	0	80	50	120	#80;	P	112	70	160	#112;	p
17	11	021	DC1 (device control 1)	49	31	061	#49;	1	81	51	121	#81;	Q	113	71	161	#113;	q
18	12	022	DC2 (device control 2)	50	32	062	#50;	2	82	52	122	#82;	R	114	72	162	#114;	r
19	13	023	DC3 (device control 3)	51	33	063	#51;	3	83	53	123	#83;	S	115	73	163	#115;	s
20	14	024	DC4 (device control 4)	52	34	064	#52;	4	84	54	124	#84;	T	116	74	164	#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	#53;	5	85	55	125	#85;	U	117	75	165	#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	#54;	6	86	56	126	#86;	V	118	76	166	#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	#55;	7	87	57	127	#87;	W	119	77	167	#119;	w
24	18	030	CAN (cancel)	56	38	070	#56;	8	88	58	130	#88;	X	120	78	170	#120;	x
25	19	031	EM (end of medium)	57	39	071	#57;	9	89	59	131	#89;	Y	121	79	171	#121;	y
26	1A	032	SUB (substitute)	58	3A	072	#58;	:	90	5A	132	#90;	Z	122	7A	172	#122;	z
27	1B	033	ESC (escape)	59	3B	073	#59;	;	91	5B	133	#91;	[	123	7B	173	#123;	{
28	1C	034	FS (file separator)	60	3C	074	#60;	<	92	5C	134	#92;	\	124	7C	174	#124;	
29	1D	035	GS (group separator)	61	3D	075	#61;	=	93	5D	135	#93;	]	125	7D	175	#125;	}
30	1E	036	RS (record separator)	62	3E	076	#62;	>	94	5E	136	#94;	^	126	7E	176	#126;	~
31	1F	037	US (unit separator)	63	3F	077	#63;	?	95	5F	137	#95;	_	127	7F	177	#127;	DEL

Source: [www.LookupTables.com](http://www.LookupTables.com)

128	Ç	144	É	160	é	176	⦿	192	Ł	208	Ł	224	α	240	≡
129	ü	145	æ	161	ı	177	⦿	193	Ł	209	ŧ	225	β	241	±
130	é	146	Æ	162	ó	178	■	194	ŧ	210	ŧ	226	Γ	242	≥
131	â	147	ô	163	ú	179		195	†	211	ŧ	227	π	243	≤
132	â	148	ô	164	û	180	†	196	—	212	Ł	228	Σ	244	ƒ
133	â	149	ò	165	Ń	181	†	197	†	213	Ł	229	σ	245	ſ
134	â	150	û	166	*	182		198	†	214	Ł	230	μ	246	•
135	ç	151	û	167	°	183	ŧ	199	†	215		231	τ	247	≈
136	ê	152	ŷ	168	ó	184	ŧ	200	Ł	216	†	232	Φ	248	•
137	ê	153	Ö	169	Ł	185		201	Ł	217	Ł	233	⊖	249	
138	ê	154	Ü	170	Ł	186		202	Ł	218	Ł	234	⊖	250	
139	ı	155	•	171	½	187	ŧ	203	ŧ	219	■	235	δ	251	√
140	ı	156	£	172	¾	188		204	†	220	■	236	∞	252	•
141	ı	157	₣	173		189		205	=	221	■	237	φ	253	•
142	Ä	158	£	174	α	190	†	206	†	222	■	238	ε	254	■
143	Ä	159	f	175	»	191	ŧ	207	Ł	223	■	239	∩	255	

Source: [www.LookupTables.com](http://www.LookupTables.com)

Font: [www.LookupTables.com](http://www.LookupTables.com)

# Nivells

## Codi Màquina

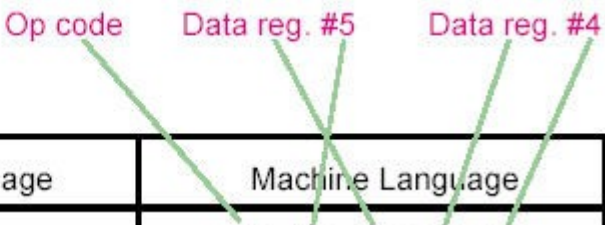
- Conjunt d'instruccions fonamentals que la màquina pot executar
- Expressada en un conjunt de 0's i 1's

## Llenguatge Ensamblador

- Equivalents alfanumèrics del llenguatge màquina
- Mnemònics més intel·ligibles

## Ensamblador

- Programa que tradueix (un-a-un) el llenguatge ensamblador a codi màquina
- El llenguatge "nadiu" del computador és el codi màquina



MC68000 Assembly Language	Machine Language
MOVE.W D4, D5	0011 101 000 000 100
ADDI.W #9, D2	00000110 01 000 010 0000 0000 0000 1001

Two Motorola MC68000 Instructions

# Nivells

Format de les **instruccions** en ensamblador: S'ajusta a les característiques de la màquina.

## Exemple

El format d'instruccions d'un processador consta de **3 camps**. El primer correspon al **OP code**, el segon al operand font i el tercer a l'altre operand font, que, a més es fa servir per depositar el resultat de l'operació

El repertori d'instruccions del procesador és de 50

Si els operands es determinen donant les adreces que ocupen en la memòria principal i aquesta pot tenir un mida màxima de 64KBytes, quin serà el format de la instrucció i la longitud dels seus camps en bits.

# Nivells

## Resposta

El Op. Code serà de 6 bits ja que amb aquests podem arribar a tenir un n° de 64 combinacions, que és superior a les 50 necessàries.

El camp dels dos operands serà de 16 bits cada un, ja que per adreçar una memòria de 64KBytes precisem de 16 bits ( $2^{16} = 65.536 = 64 \text{ KB}$ )

Per tant el format de la instrucció serà de 38 bits

Hi ha altres possibilitats, que es veuen al capítol 3

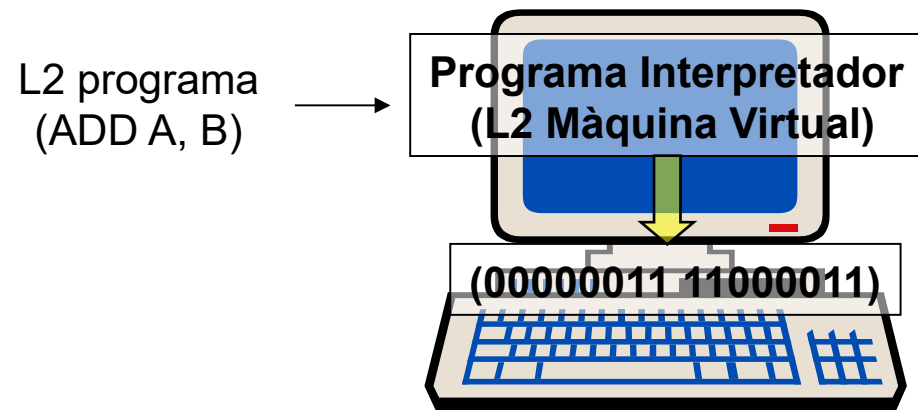
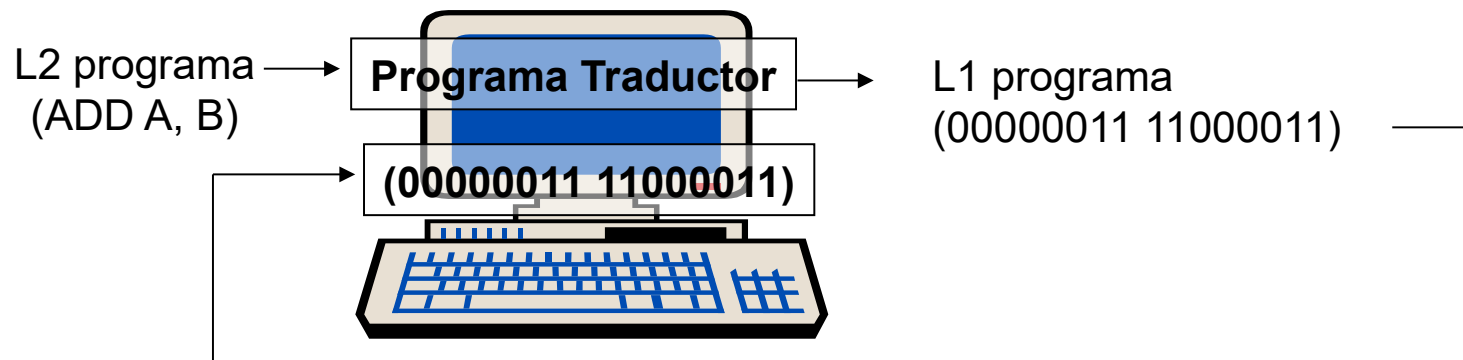
# Traducció - Interpretació

Instrucció L1 :

00000011 11000011

Instrucció Equivalent L2 :

ADD A, B



# Nivell Sistema Operatiu

- Un Sistema Operatiu és un *programa* que proporciona un entorn en el que l'usuari pot executar altres programes de manera convenient i eficient.
- És responsable de controlar els recursos del sistema (en temps i espai), assignant recursos als programes d'usuari i monitorant la integritat del sistema
- Proporciona serveis a programes per facilitar la tasca del programador. Aquests serveis poden ser invocats per el programa d'usuari a través de crides al sistema (*system calls*)
- La majoria del codi generat per nivells superiors és passat directament al nivell llenguatge màquina

# Llenguatges Alt-Nivell

- Llenguatges Compilats: Pascal, C, C++, Fortran, Java
- Llenguatges Interpretats: Perl, Python, JavaScript, Java
- Amaguen al programador els detalls arquitecturals de baix nivell
- Els llenguatges d'alt nivell són generalment independents de la màquina

Source code written in a language a human can **easily** understand

```
X := 5; Y := 3;  
Z := 4*X+Y*3;
```

Compiled into...

Source code written in a language a computer can easily understand

```
mov eax, 5  
mov edx, 3  
shl eax, 2  
lea edx, [edx+edx*2]  
add eax, edx
```

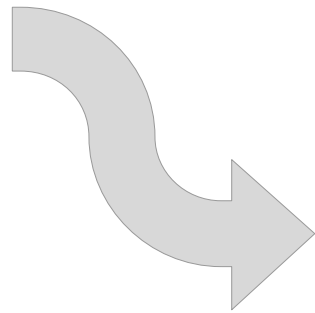
Assembled into...

Executable program (i.e., machine code directly read by the CPU hardware)

```
10111000 00000101 00000000 00000000 00000000  
10111010 00000011 00000000 00000000 00000000  
11000001 11100000 00000010  
10001101 00010100 01010010  
00000011 11000010
```

# Llenguatges Alt-Nivell

```
for (int i=0;i<N_MAX;i++){  
    double b = i+CONSTANT;  
    ...  
}
```



```
SUB R1,R1,R1 ;# sub font1, font2, destí  
SUB R2,R2,R2  
SUB R3,R3,R3  
SUB R4,R4,R4  
ADDI R4,CONSTANT,R4  
for_loop:  
ADDI R1, N_MAX, R1  
loop:  
    ADD R4,R2,R2  
    ...  
    ADDI R2,1,R2  
    SUBI R1,1,R1  
BG loop
```



# Estructura i Funció

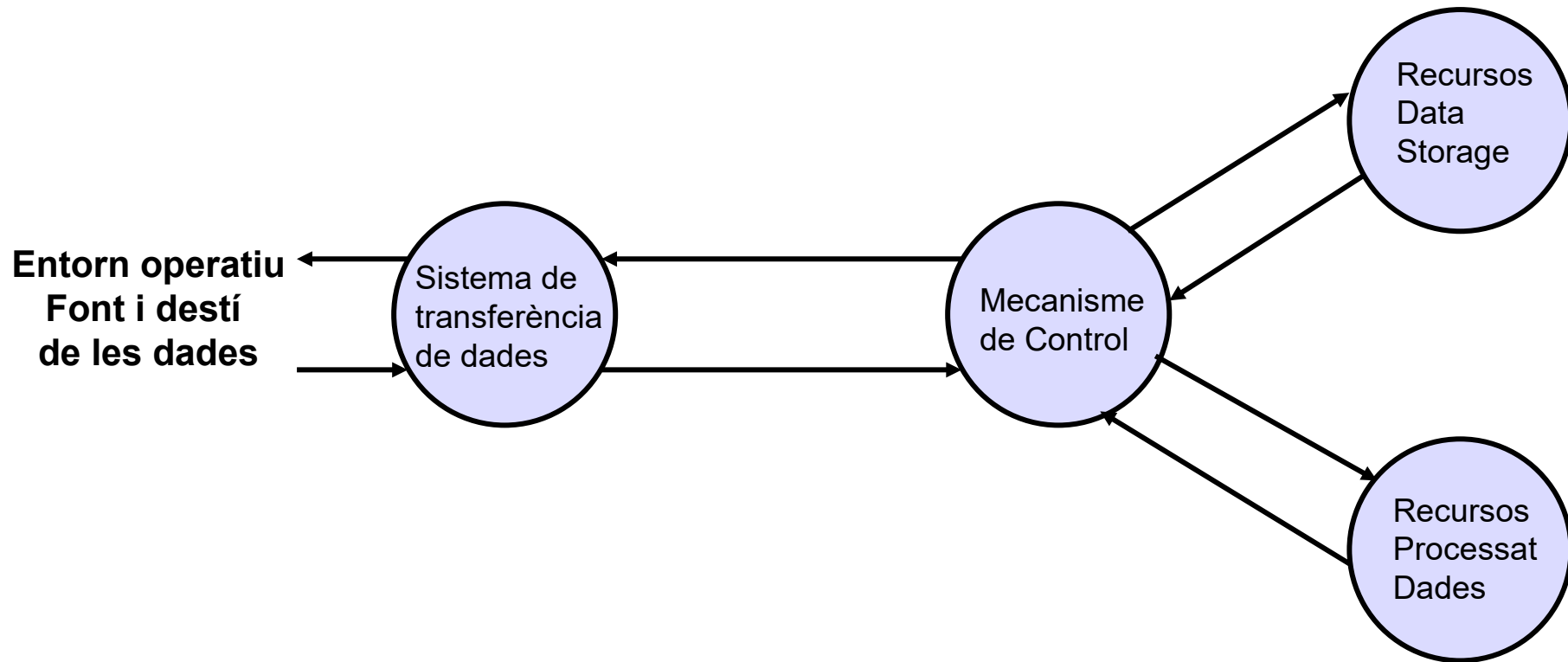
- A cada nivell el dissenyador ha de considerar
  - **Estructura:** La manera amb que es relacionen uns components amb els altres.
  - **Funció:** L'operació dels components individuals com a part de l'estructura

## Funció

- Les funcions d'un computador són:
  - Processat de dades
  - Emmagatzematge de dades
  - Moviment de dades
  - Control

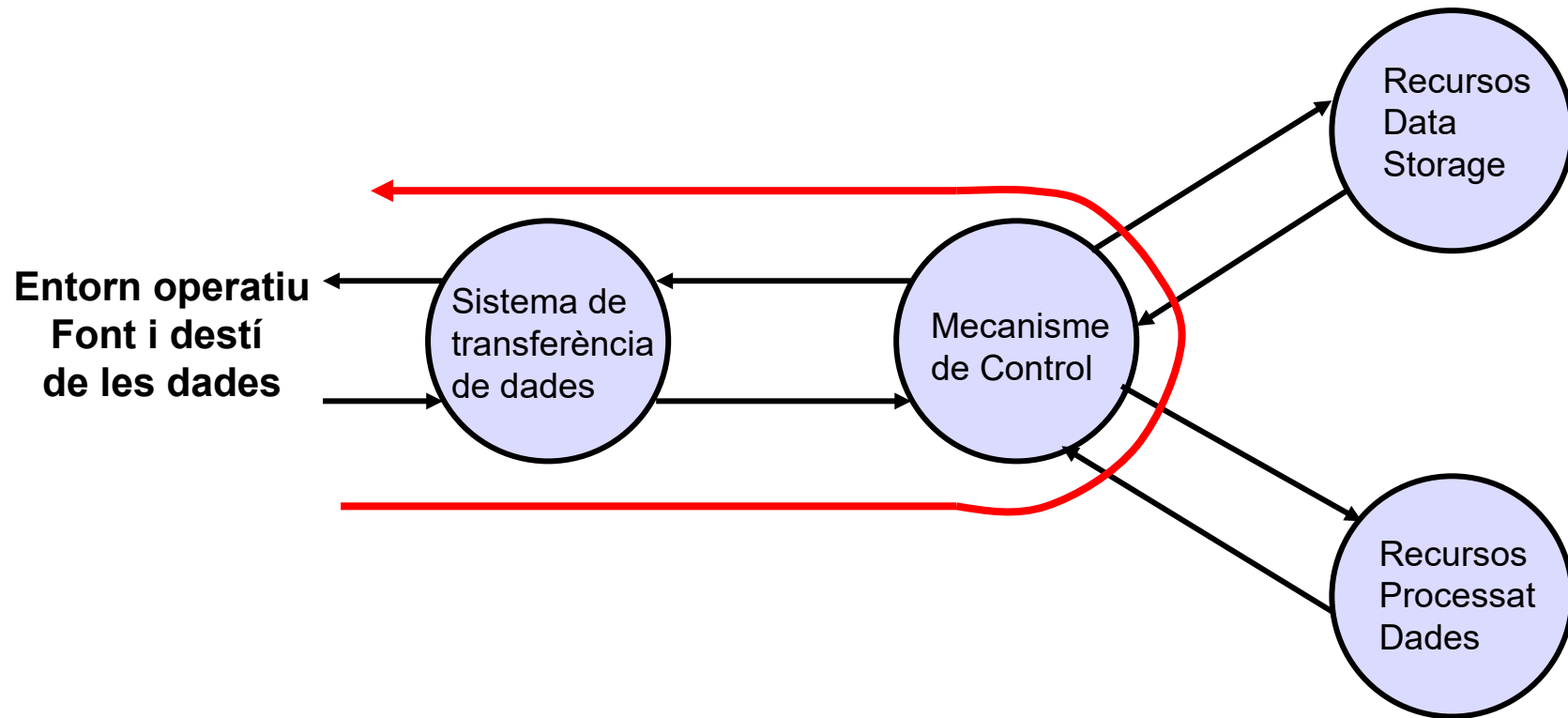
# Visió Funcional

- Visió funcional d'un computador



# Operacions (1)

- Moviment o transferència de dades
  - P.ex. Teclat a pantalla



# Moviment o transferència de dades

- Exemple de moviment de dades

1) De teclat a memòria i de memòria a pantalla  
(Llenguatge ensamblador i8085)

- IN 04h      (captura de dades DES DE l'adreça 04h)

IN 04h # guardo contingut de disp. Localitzat en l'adreça 04h en registre A

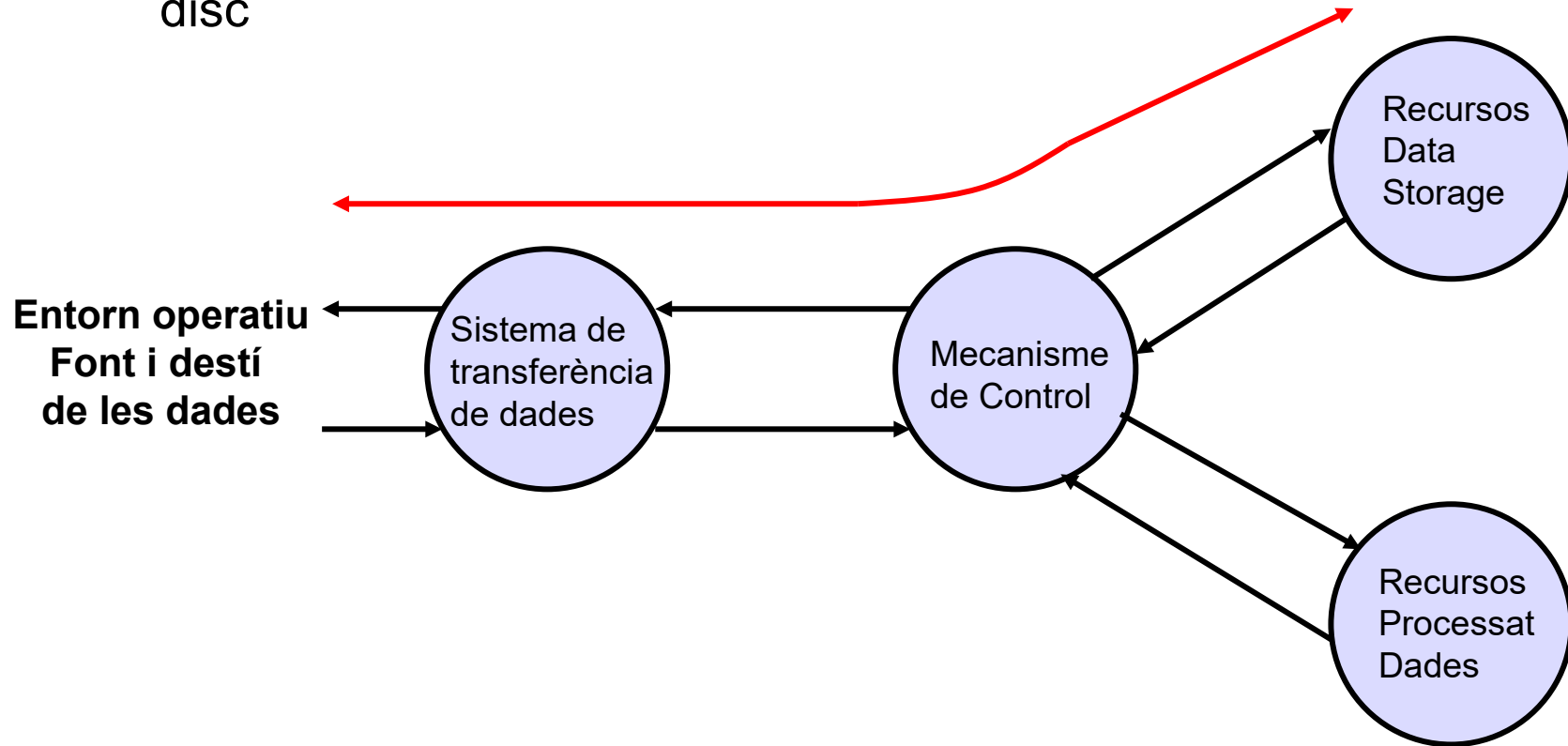
STA Memoria # guardo contingut registre A en la posició de memòria Memoria

- OUT 05h    (sortida de dades CAP A l'adreça 05h)

?

# Operacions (2)

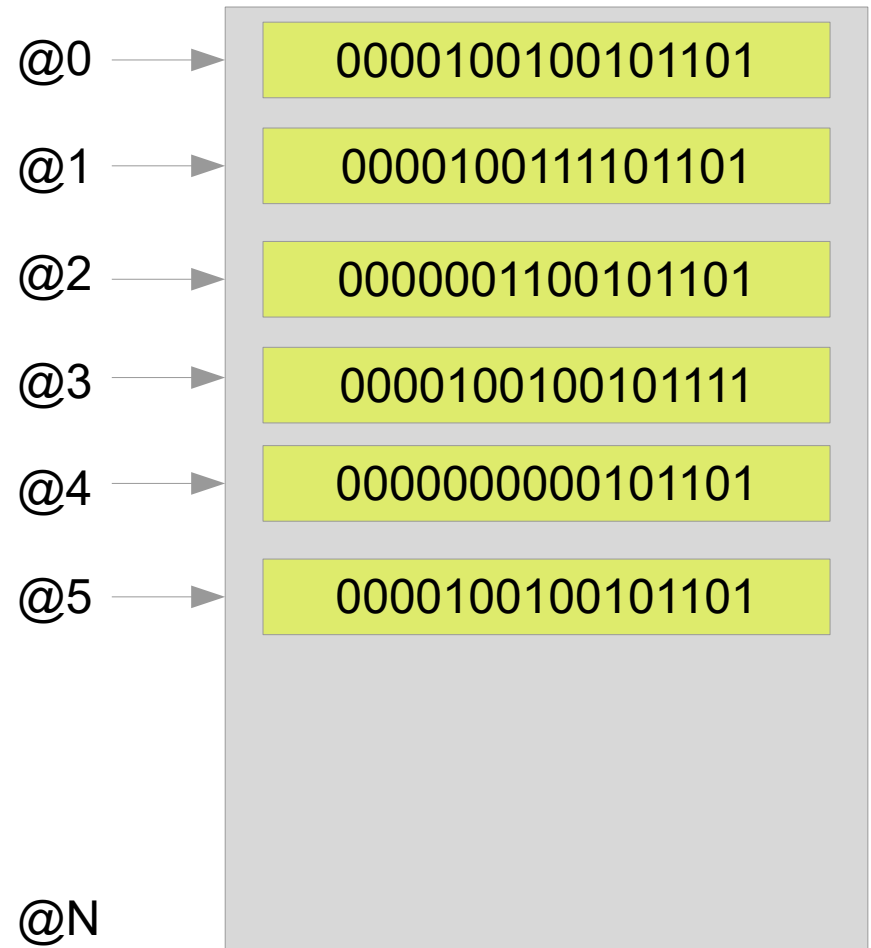
- Emmagatzematge (Data Storage)
  - P.ex. Dades transferides des d'un entorn extern (internet) a disc



# Emmagatzematge (Data Storage)

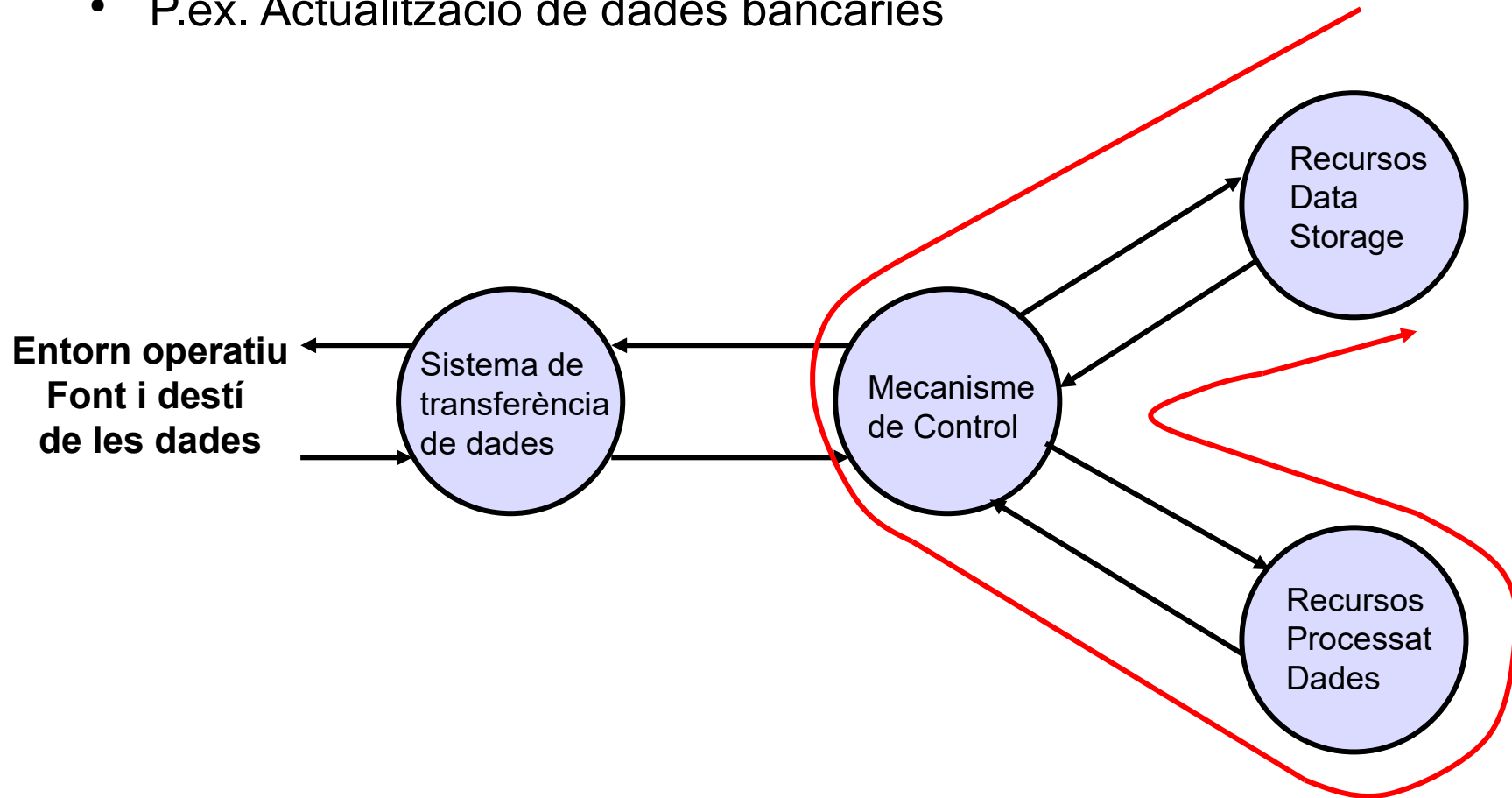
- 1) De memòria a CPU
  - LOAD offset(R1), R2
- 2) De CPU a memòria
  - STORE R1, offset(R2)

Ex. [R1] = 0, offset = 2



# Operacions (3)

- Processat de dades emmagatzematge
  - P.ex. Actualització de dades bancàries



# Processat i emmagatzematge

- Exemple

LOAD 3(R0), R1

LOAD 4(R0), R2

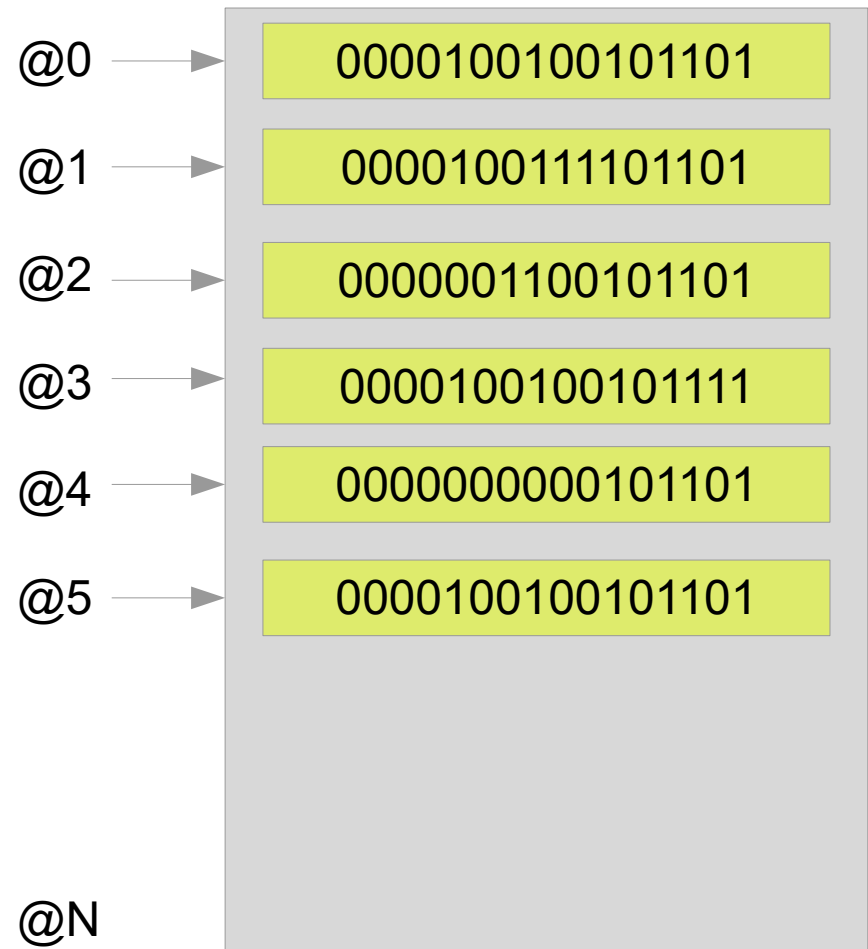
SUB R3,R3,R3

loop: ADD R1,R3,R3

    SUBI R2, 1, R2

BG loop

STORE R3, 5(R0)

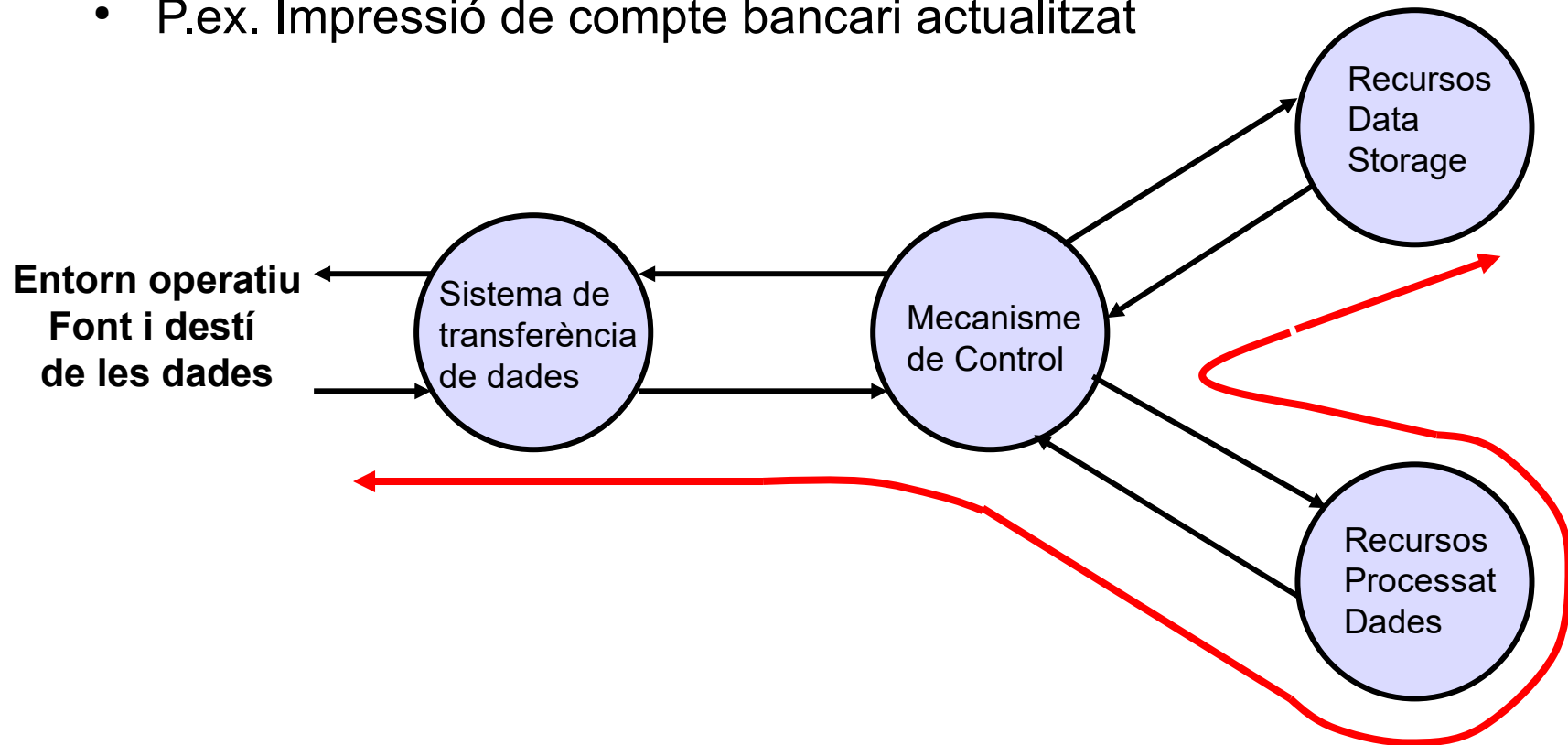




# Operacions (4)

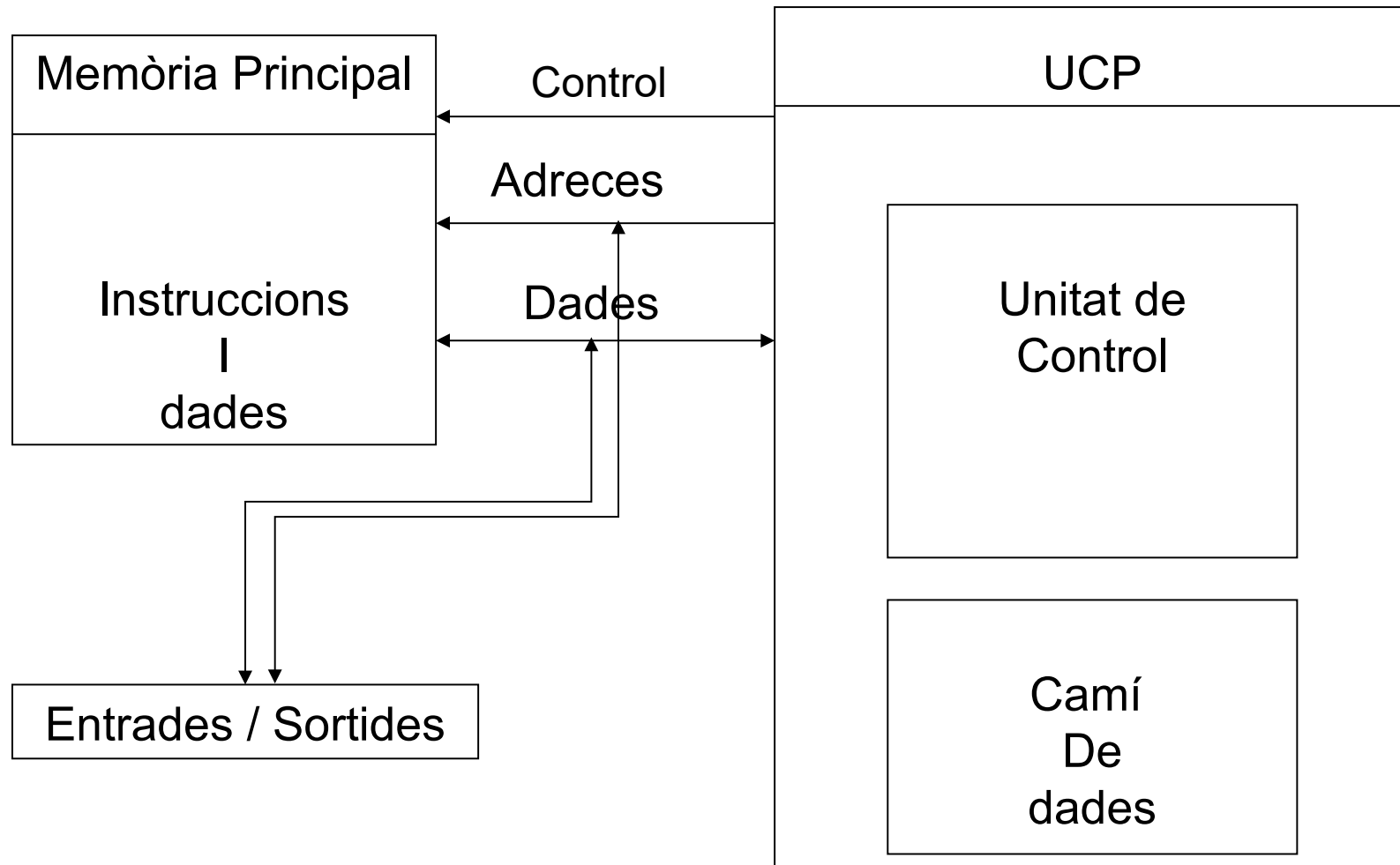
- Processat des d'emmagatzematge a Entrada/Sortida

- P.ex. Impressió de compte bancari actualitzat



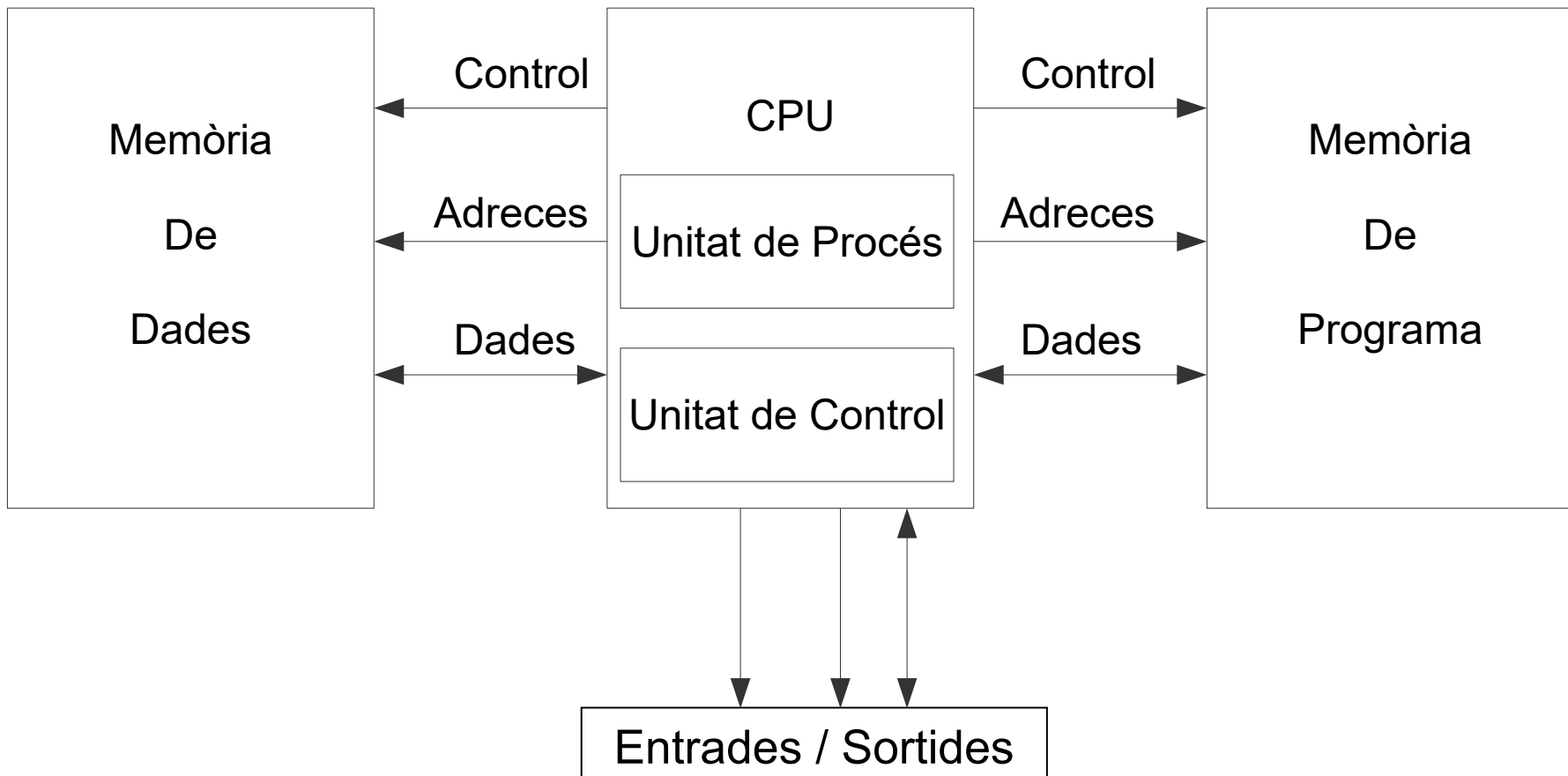
# Nivells

Estructura simplificada del computador proposada per Von Neumann



## Nivells

Estructura Harvard. Usualment utilitzada en microcontrol·ladors i sistemes empotrats

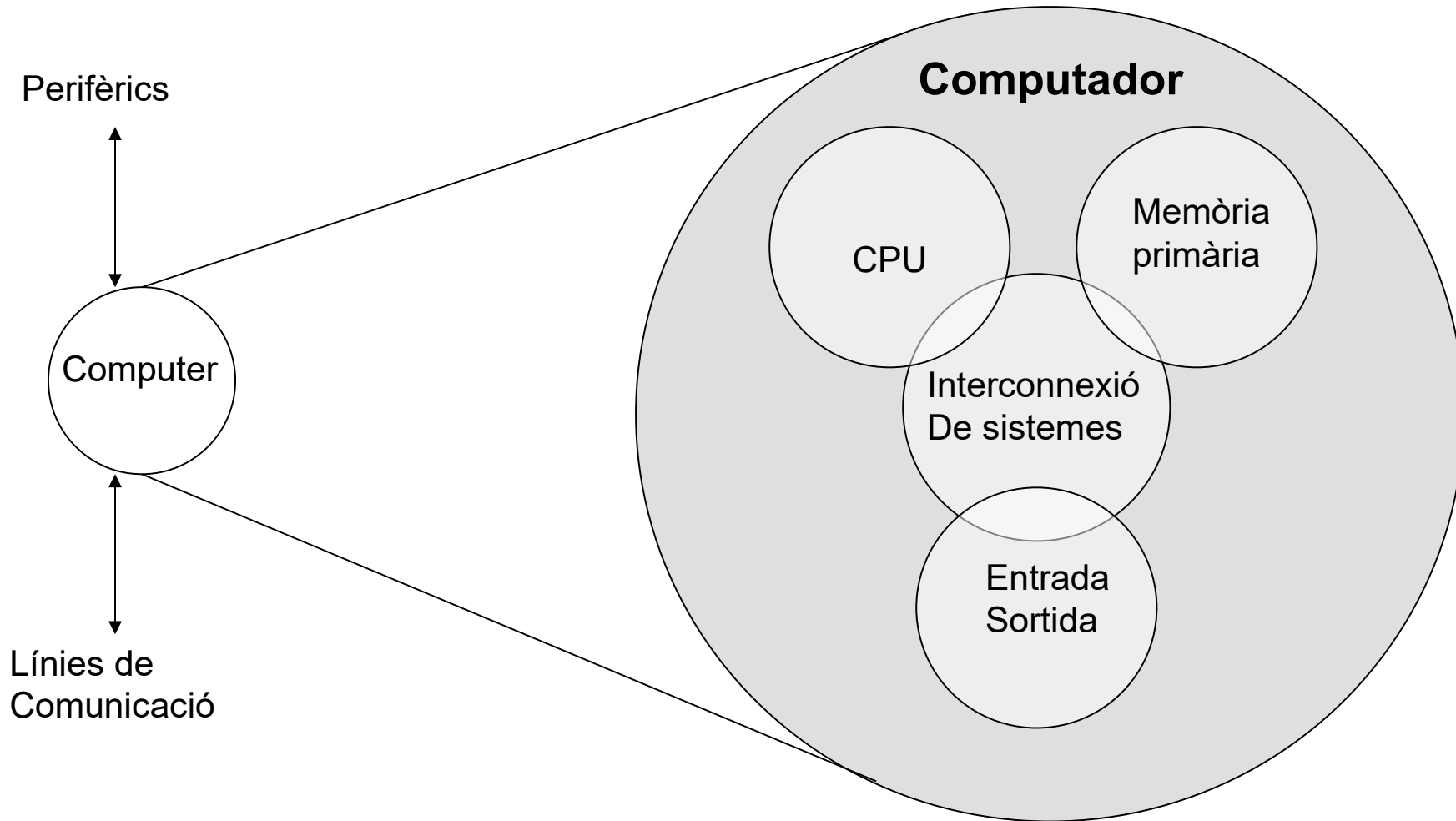


# Estructura

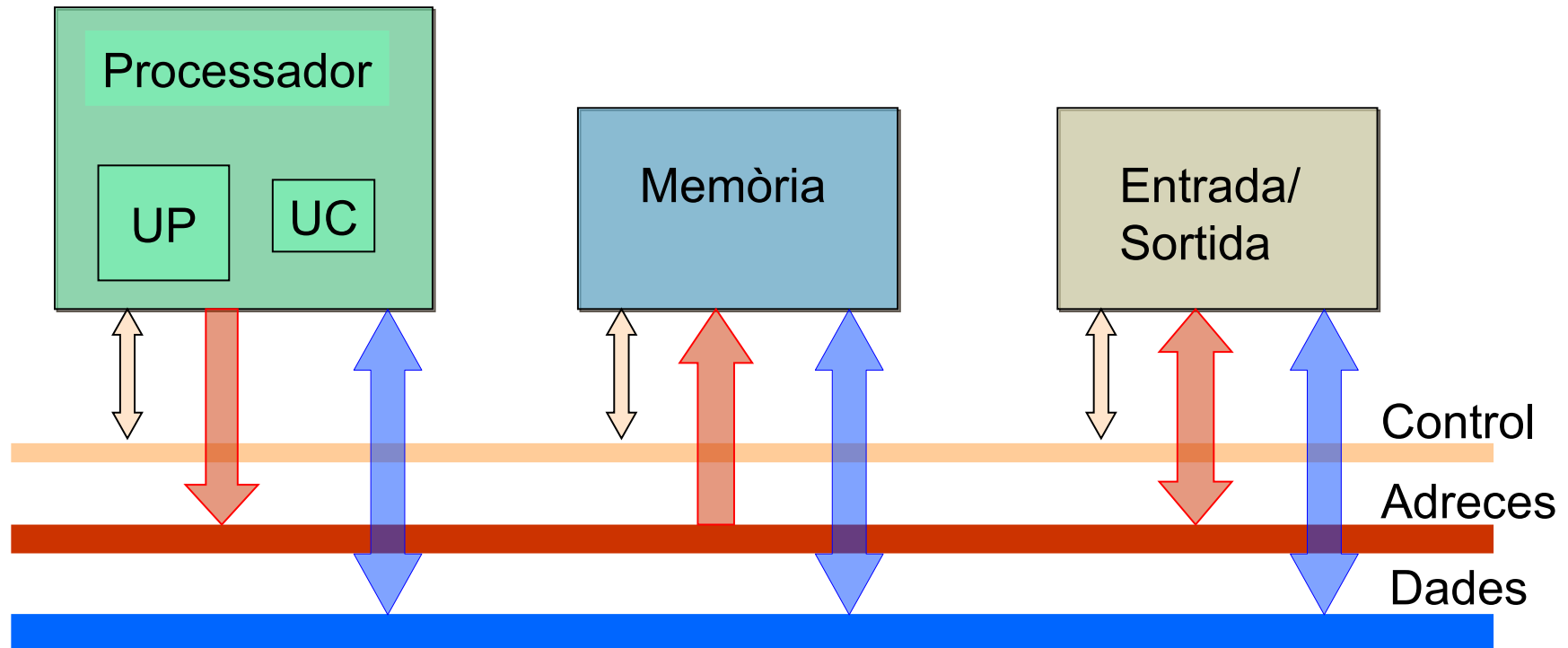
- Principals Components d'un Computador
  - **Unitat Central de Procés (CPU)**  
Controla l'operació del computador i fa el processat de dades
  - **Memòria primària**  
Guarda dades
  - **Entrada / Sortida E/S**  
Mou dades entre el computador i l'entorn extern
  - **Sistema de connexió (BUS)**  
Mecanisme que proporciona comunicacions entre els components del sistema,

BUS= conjunt de línies agrupades que condueixen senyals relacionats

# Estructura – Nivell superior



# BUS Genèric de sistema



Bus de sistema = constituït per 3 busos:

Dades  
Adreces  
Control

# BUS Genèric de sistema

Bus de sistema:

- Constituït per 50 – 100 línies. Cada una amb la seva funció particular.

- **Bus de dades:** Proporcionen un camí per transmetre dades entre els mòduls del sistema. Normalment consta de 8, 16 o 32 línies diferents (amplada del bus). Això determina el nº de bits a transmetre en un interval de temps.

- 1.- L'amplada del bus és clau per determinar les prestacions del conjunt del sistema

Ex: Amplada de bus de 8 bits, instruccions de 16 bits

# BUS Genèric de sistema

-**Bus d'Adreces:** S'utilitza per designar la font o el destí de les dades. P.E. Si la CPU desitja llegir una paraula (8, 16, o 32 bits) de dades de la memòria, posarà la direcció de la paraula al bus de direcció.

- i) L'amplada del bus d'adreces determina la màxima capacitat de memòria possible del sistema.
- ii) Les línies d'adreces també s'utilitzen per adreçar els ports d'E/S.
- iii) Els bits d'ordre més alt s'utilitzen per seleccionar una posició de memòria o un port E/S dintre del mòdul.

Ex: BUS de 8bits. La direcció 01111111 o inferiors fan referència a posicions dintre d'un mòdul de memòria. La direcció 10000000 i superiors designen dispositius connectats a un mòdul E/S,



# BUS Genèric de sistema

-**Bus de control.** S'utilitza per controlar l'accés i l'ús de les línies de dades i de adreces. Això es fa ja que les línies de dades i d'adreces són compartides per tots els components, i per tant ha d'existir una forma de controlar el seu ús.

- i) Transmissió d'ordres: Especifiquen les operacions a realitzar
- ii) Transmissió d'informació de temporització entre mòduls del sistema, indicant la validesa de les dades i les adreces.

Ex. Línies de control típiques:

- . Escriptura en Memòria
- . Escriptura de E/S
- . Lectura de E/S
- . Petició de Bus
- . Cesió de Bus
- ....

# BUS Genèric de sistema

Funcionament del BUS:

Si un mòdul desitja enviar una dada a un altre mòdul farà dues coses:

- 1.- Obtenir l'ús del bus
- 2.- Transferir la dada a través del bus

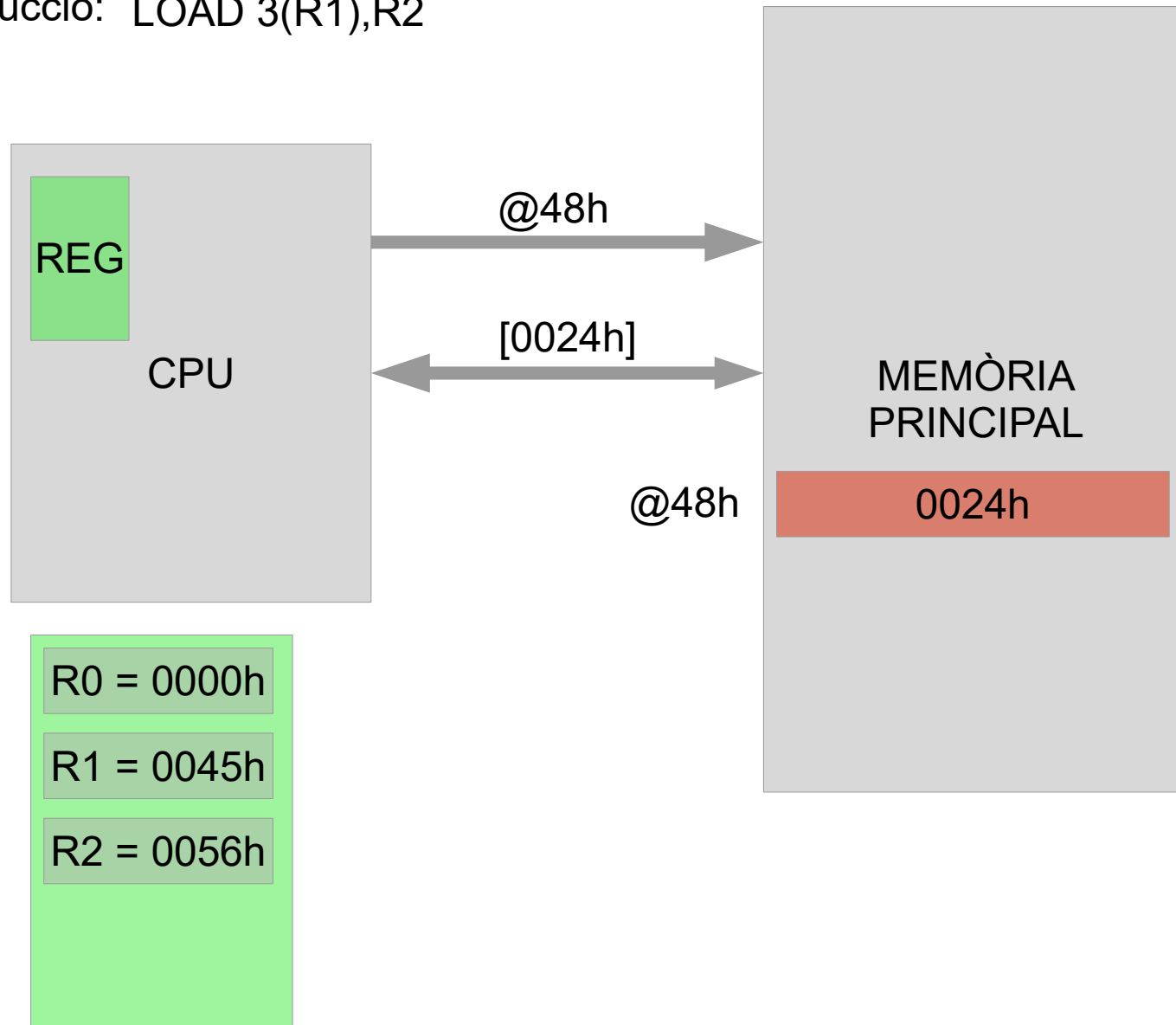
Si un mòdul desitja demanar una dada un altre mòdul haurà de:

- 1.- Obtenir l'ús del bus
- 2.- Transferir la petició a l'altre mòdul mitjançant les línies de control i adreces apropiades

# BUS Genèric de sistema

Exemple d'ús dels busos:

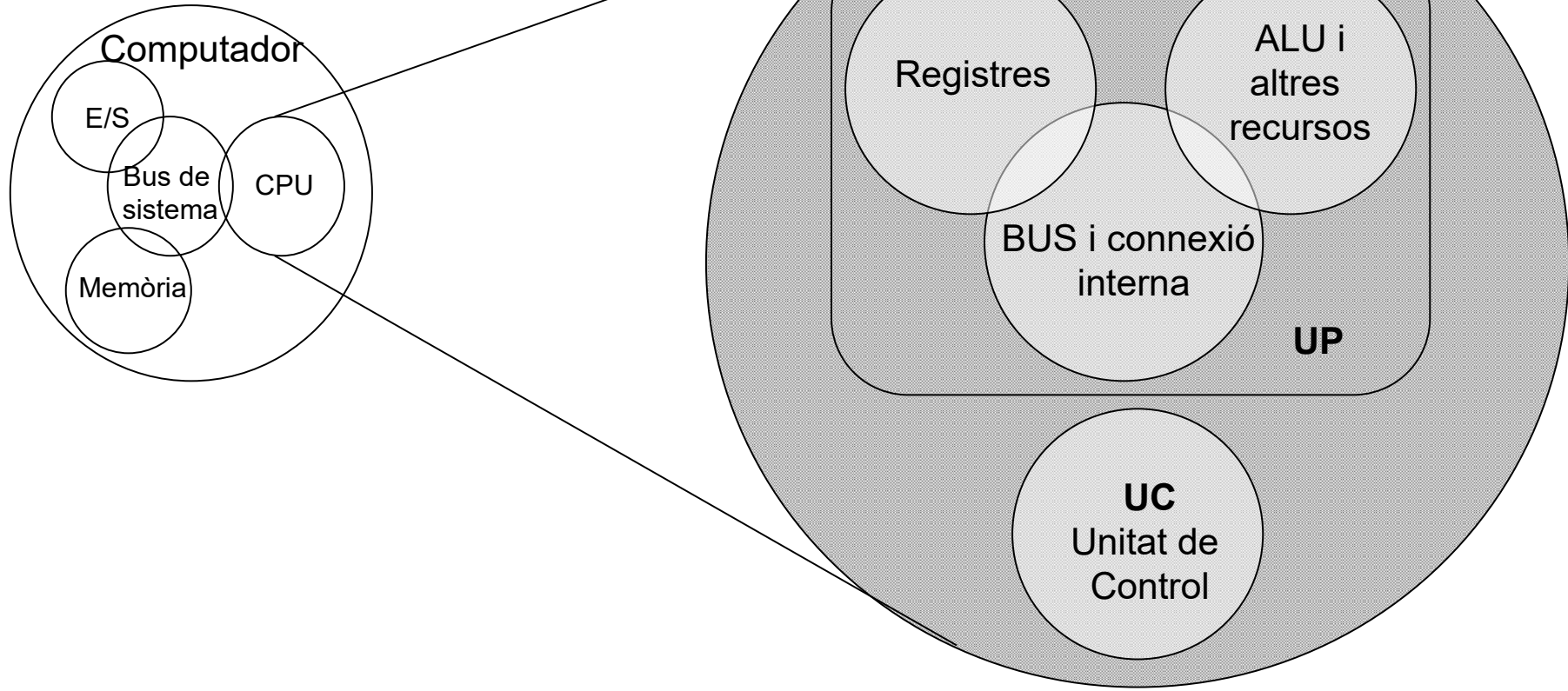
Instrucció: LOAD 3(R1),R2



# Estructura - La CPU

Principals components de la CPU

- UNITAT DE CONTROL (UC)
- UNITAT DE PROCÉS (UP)



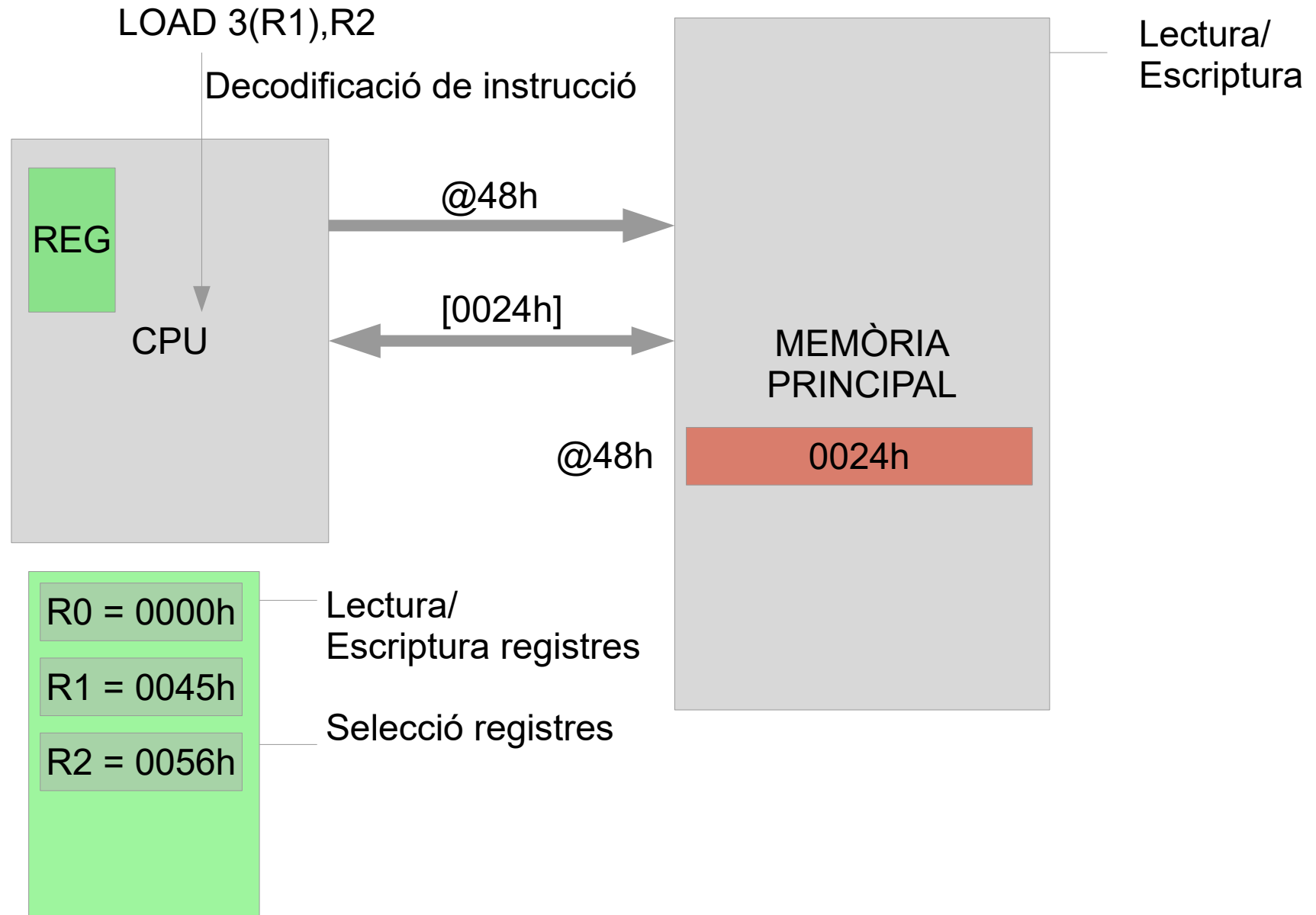
# Estructura - Unitat de procés (UP)

- ALU i altres recursos:
  - Unitat aritmètico-lògica
    - Basada en sumador
      - Suma, resta, complementació,  $\times 2$ ,  $/2$
      - AND, OR, XOR,... (bit-a-bit)
  - Multiplicador, Desplaçador, LUT
- REGISTRES
  - Propòsit específic:
    - IR, PC, SP, Status, AR, DR, AC
  - Propòsit general
    - Conjunt de registres (dual port)
- BUS i connexions internes
  - DADES; ADRECES; CONTROL
  - Diferents estructures de BUS dades

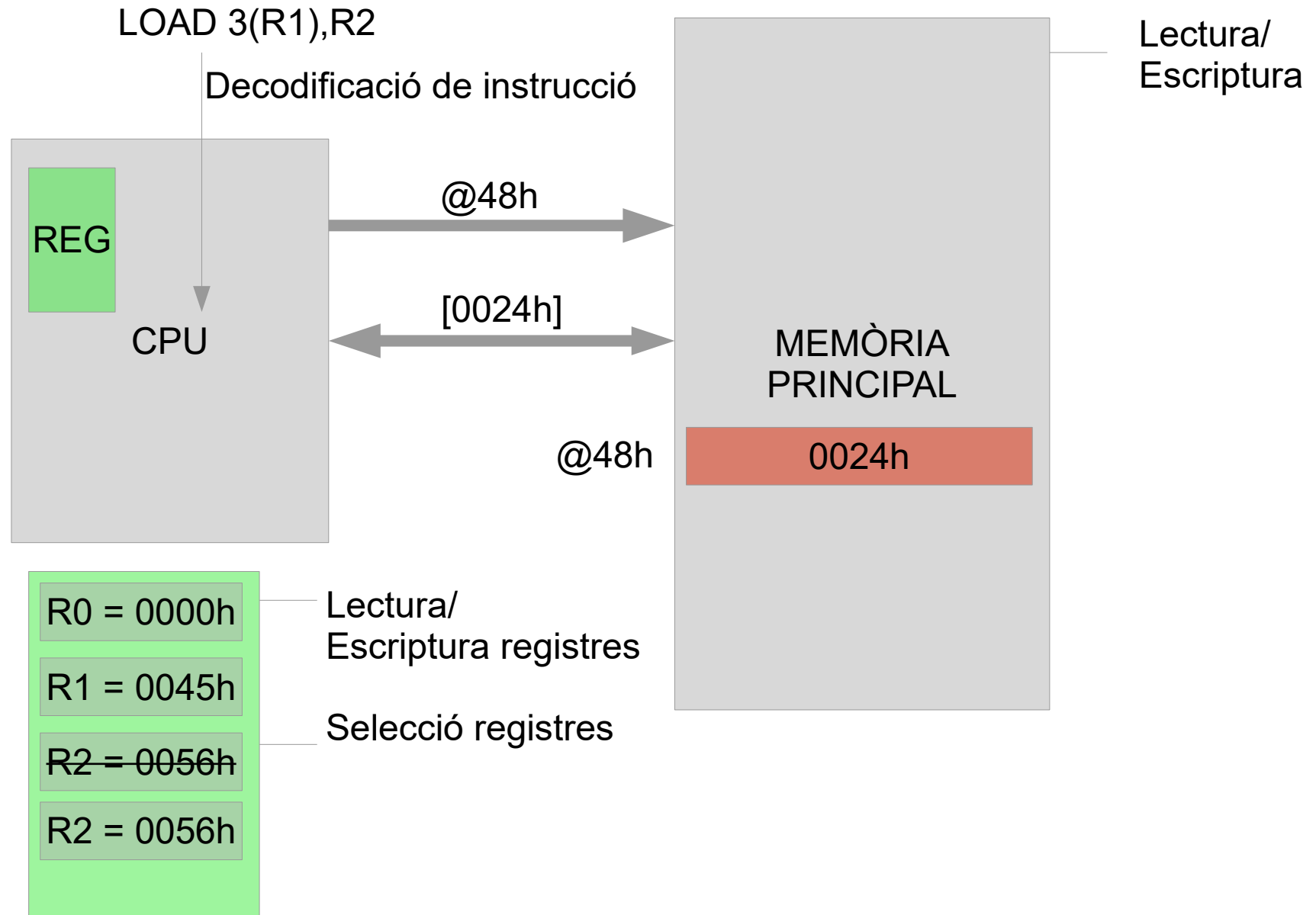
# Estructura – Unitat de Control (UC)

- Unitat de Control cablejada (*hardwired*)
  - Màquina d'estats fixa.  
Diferents estratègies de disseny
- Unitat de Control Microprogramada
  - La unitat de control és en si un petit computador.
  - Una instrucció del processador és implementada amb un microprograma amb determinat nombre de microinstruccions
  - Lògica seqüencial – Control de l'ordre dels events
    - Microprograma
    - Memòria del microprograma

# Estructura busos. Accés MP-CPU

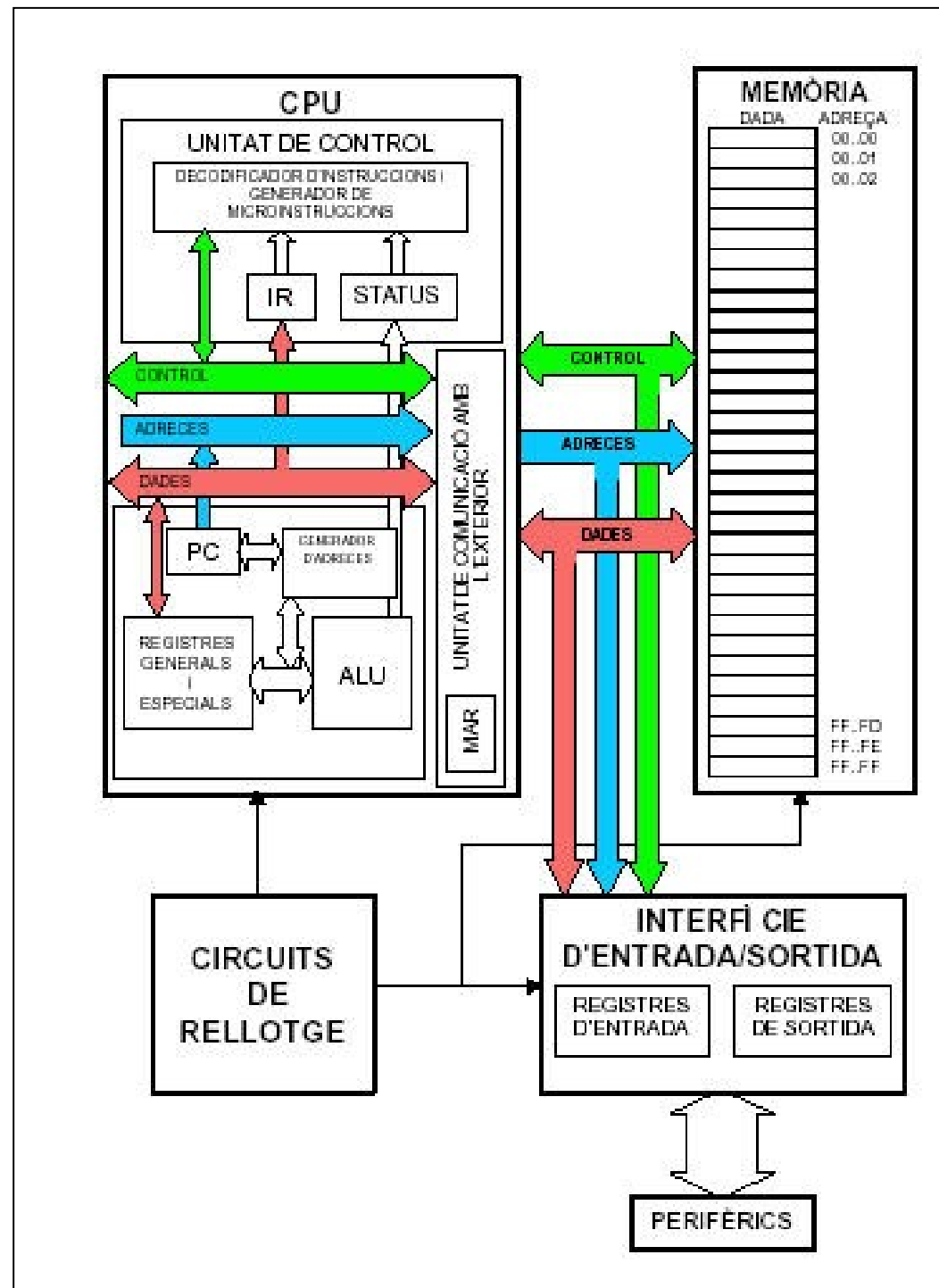


# Estructura busos. Accés MP-CPU





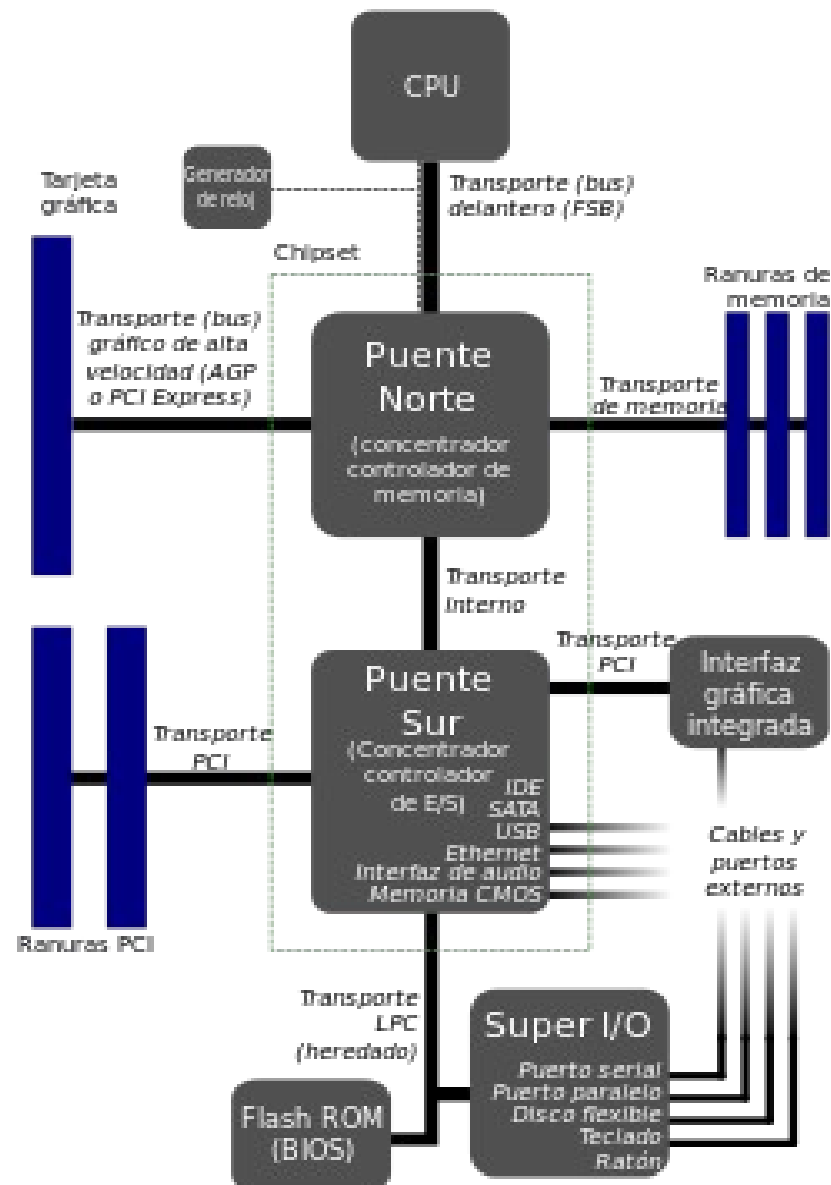
# Diagrama de blocs General d'un Computador



# Placa Base: Chipset

North Bridge: Controla l'accés a CPU, MP i AGP

- Conecta la CPU amb MP
- Conecta la CPU amb AGP o bé
- Conecta la CPU amb PCI Express



\*Font: Puente Norte - Wikipedia

# North Bridge

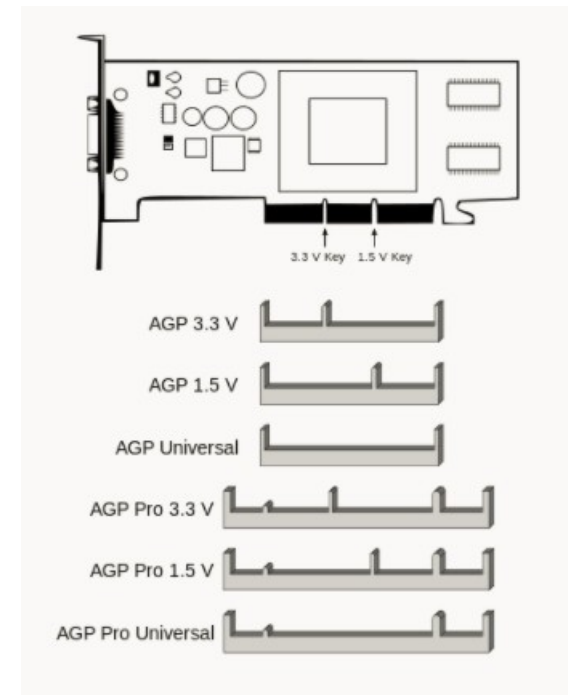
- PCI Express és un desenvolupament del bus PCI molt més ràpid
- Es basa en el bus PCI, està pensat per fer-se servir com a bus local.
- Cada ranura d'expansió porta 1, 2, 4, 8 ó 16 carrils de dades entre la placa base i les targetes connectades. Pot assolir amplex de banda de 500MB/s per cada canal → en el cas de fer servir x16 podem arribar a 8GB/s en cada adreça èr a PCIE 2.x
- 8 carrils tenen un ample de banda comparable a la versió més ràpida de AGP

\*Font: PCI Express - Wikipedia



# North Bridge

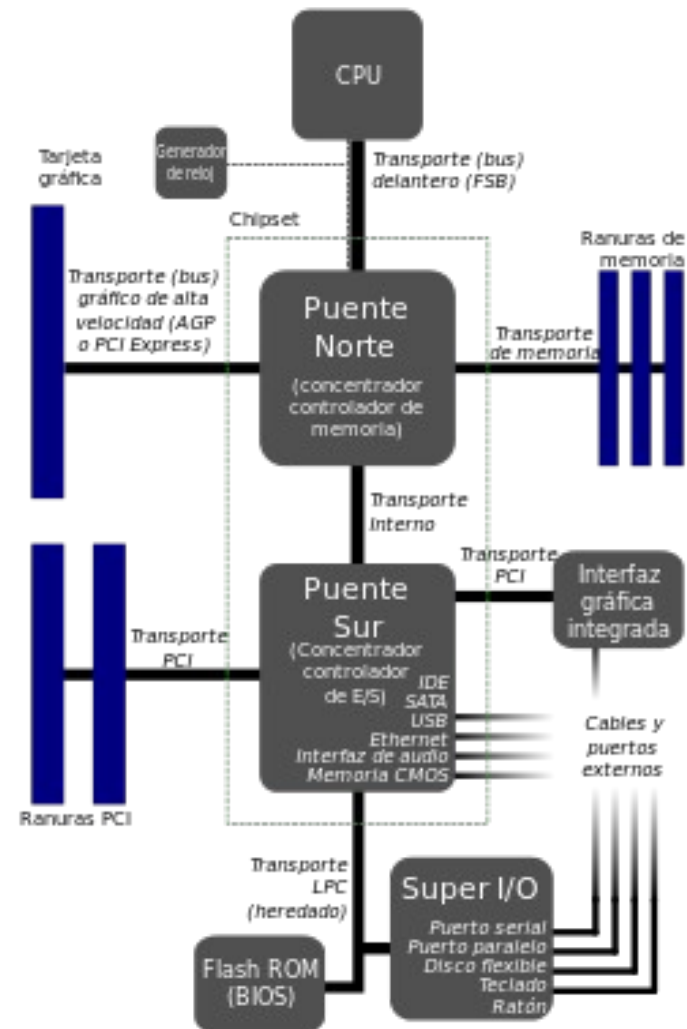
- AGP (Accelerated Graphics Port) és una especificació de bus que proporciona una connexió directa entre l'adaptador de gràfics i la memòria. És un port → Només permet connectar un dispositiu
- Conté la majoria dels senyals del bus PCI + agregats



# Placa Base. Chipset

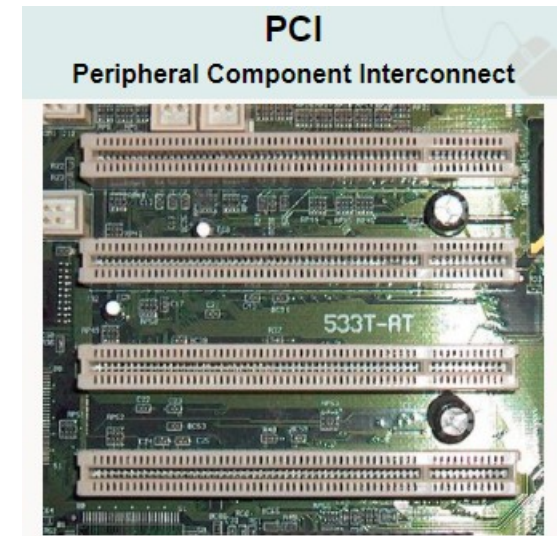
La funcionalitat que trobem als southbridges actuals inclou:

- Bus PCI
- Bus ISA
- Bus SPI
- System Management Bus ( SMBus )
- Controlador DMA
- Controlador de Interrupcions
- Controlador IDE (SATA o PATA)
- Pont LPC
- Real Time Clock
- Administració de potencia eléctrica APM i ACPI
- BIOS
- Interfaç de so AC97 o HD Audio.



# South Bridge

- Peripheral Component Interconnect o PCI és un bus estàndard de computadores per connectar dispositius perifèrics directament a la placa base
- Molt comú en Pcs i Labtops on ha substituït al bus ISA



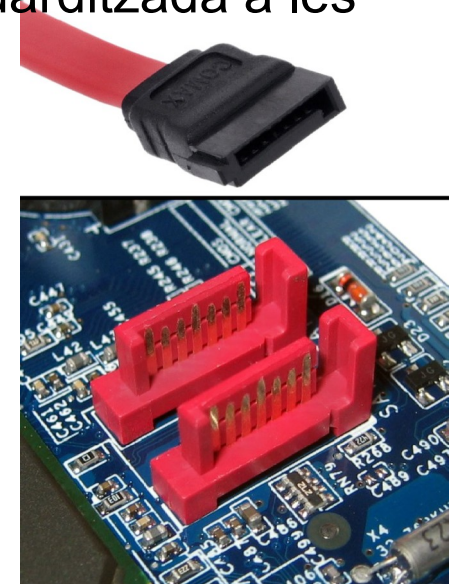
# South Bridge

- Industry Standard Architecture (ISA). Dissenyat per connectar targetes d'ampliació a la placa base. Baixa velocitat. Substituït per PCI



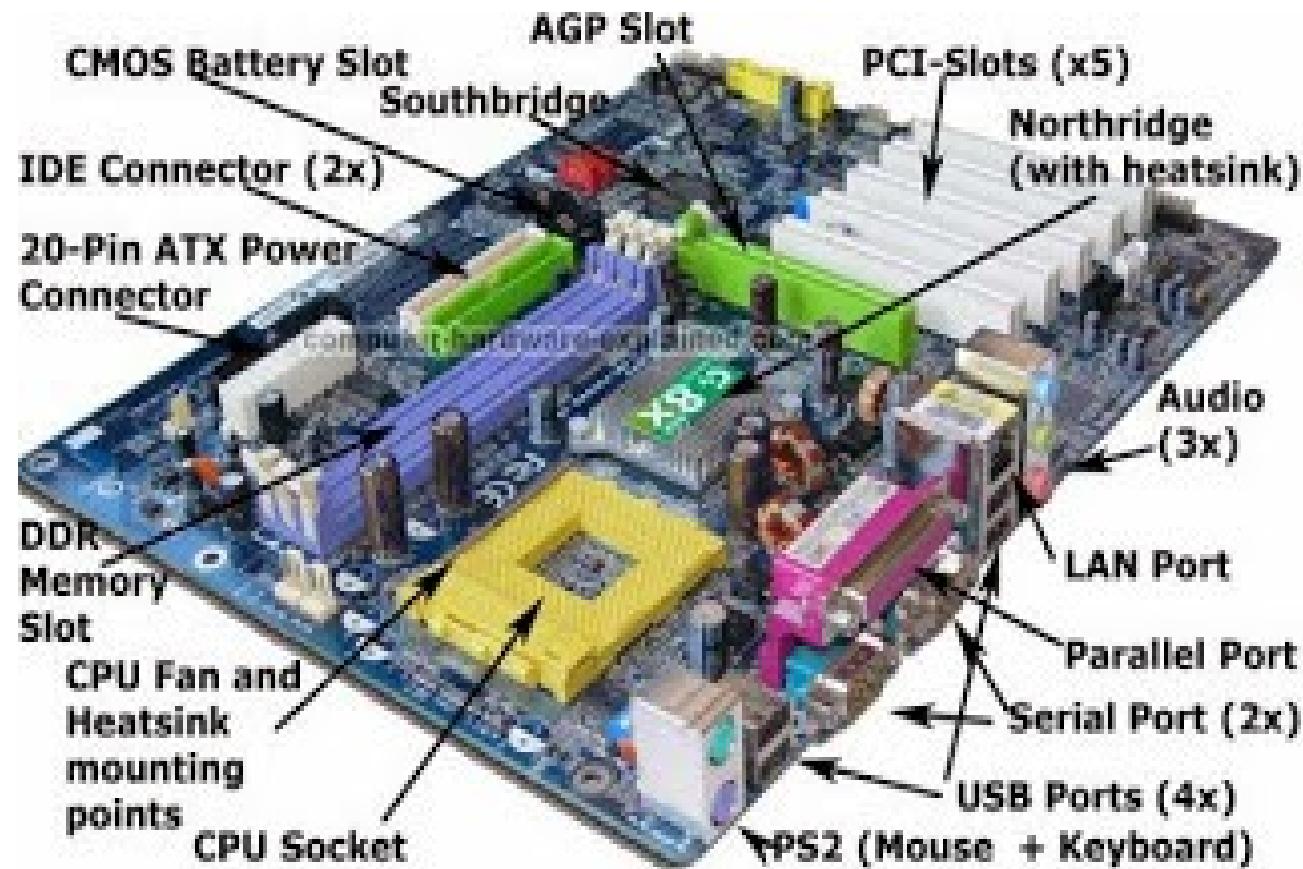
# South Bridge

- Serial ATA o S-ATA (acrònim de Serial Advanced Technology Attachment) és una interfície de transferència de dades entre la placa mare i alguns dispositius d'emmagatzematge, com pot ésser el disc dur, o bé d'altres dispositius d'altres prestacions que encara s'estan desenvolupant. Serial ATA substitueix la tradicional Parallel ATA o P-ATA (estàndard que també és conegut com a IDE o ATA). El S-ATA proporciona velocitats més altes, més aprofitament quan hi ha diversos discos, més longitud de cable de transmissió de dades i capacitat per a connectar discos en calent (amb l'ordinador encès).
- Actualment és una interfície extensament acceptada i estandarditzada a les plaques mares de PC.





# Placa Base: Exemple



# Placa Base: Exemple

