|  |  |
| --- | --- |
|  |  |

**Sistemes informàtics**

Arquitectura de computadors

# 

## 1. Estructura i components d’un sistema informàtic

### 1.1. Què és un sistema informàtic

## Un sistema informàtic (SI) és un sistema que permet **emmagatzemar** i **processar informació**; és el conjunt de parts interrelacionades: **maquinari**, **programari** i **personal informàtic**.

## La **informació** és el resultat de la manipulació de les **dades**, treballant-les i ordenant-les amb la finalitat de produir un coneixement.

## El **maquinari** inclou computadores o qualsevol tipus de dispositiu electrònic, microprocessadors, memòria, sistemes d’emmagatzematge extern, etc.

## El **programari** inclou al sistema operatiu, firmware i aplicacions.

## 

### 1.2. El sistema binari

El sistema binari o de base 2 és el sistema numèric posicional que utilitza dos símbols per representar els nombres. Els símbols són {0,1} i a cadascun d’ells se’ls anomena xifra binària o bit i es defineix com la unitat més petita de representació de la informació.

Les agrupacions de bits, depenent de la seva longitud, reben un nom:

|  |  |
| --- | --- |
| 4 bits | nibble |
| 8 bits | byte |
| 16 bits | mitja paraula |
| 32 bits | paraula |
| 64 bits | doble paraula |

El bit i el byte són unitats de mesura de la quantitat d’informació. Per tant, s’utilitzen els mateixos prefixes que per qualsevol altre unitat per descriure els seus múltiples.

Fa alguns anys, aquesta unitat era suficient per mesurar la quantitat d’informació que hi havia en aquells moments, però avui resulta massa petita per als grans volums d’informació que es manipula i s’utilitzen prefixos per anomenar als múltiples del byte. S’utilitzen prefixos del SI o bé els prefixos binaris (IEC 60027-2).

En la pràctica popular, els prefixos binaris corresponen a nombres similars als factors indicats en el SI. Els primers són potències amb base 2, mentre que els prefixos del SI són potències amb base 10.

Aquesta diferència pot donar lloc a confusió a l’hora de mesurar quantitats de dades. Per tal d’evitar-ho, l’any 1998 la IEC va desenvolupar un estàndard on es varen definir unitats per a aquests prefixos binaris.

A continuació es mostren els dos sistemes de mesura:

|  |  |  |  |
| --- | --- | --- | --- |
| **Prefix del SI (SI)** | | **Prefix binari (IEC 60027-2)** | |
| kilobyte kB | 103 bytes | kibibyte KiB | 210 bytes |
| megabyte MB | 106 bytes | mebibyte MiB | 220 bytes |
| gigabyte GB | 109 bytes | gibibyte GiB | 230 bytes |
| terabyte TB | 1012 bytes | tebibyte TiB | 240 bytes |
| petabyte PB | 1015 bytes | pebibyte PiB | 250 bytes |

En el món informàtic, el qual ja s’ha estès cap a la vida quotidiana, és molt habitual utilitzar els prefixos del SI quan realment haurien de fer servir els prefixos de la IEC. Per exemple, ens podem trobar especificacions tècniques que parlen de GB (gigabytes) quan realment haurien de dir GiB (gibibytes). Això passa perquè són prefixos de mesura molt similars. Fixeu-vos que 1 megabyte (1 MB) equival a 1.000.000 de bytes (106), i 1 mebibyte (1 MiB) equival a 1.048.576 bytes (220).

A mida que els prefixos augmenten (Gibi, Tebi,…), també s’incrementa la diferència entre tots dos sistemes. Així doncs cal parar atenció a la utilització correcta de les unitats.

La capacitat d’emmagatzematge és el camp d’aplicació habitual dels prefixos binaris i de les mesures informàtiques a partir del byte. En el camp de les mesures de les **velocitats de les comunicacions** és més comuna la utilització de prefixos del SI i d’unitats a partir del bit. Així doncs us podeu trobar amb la velocitat d’una xarxa indicada a 100 megabits per segon (100 Mbps).

|  |  |
| --- | --- |
|  | En general quan es treballa en Bytes les equivalències entre els prefixes són en potències de 2.  Quan es treballa en bits s'acostuma a treballar en potències de 10 |

|  |  |
| --- | --- |
| 1 kilobit (Kbit) | 1000 bits |
| 1 megabit (Mbit) | 1000 Kbit |
| 1 gigabit (Gbit) | 1000 Mbit |
| 1 terabit (Tbit) | 1000 Tbit |
| 1 petabit (PB) | 1000 Tbit |

### 

#### Exercicis

1. Quin avantatge creus que pot tenir treballar amb un sistema de més bits de registres interns? Pensa com sumaries 23 + 8 en un sistema que treballi amb CPU de 4 bits i com ho faria amb una CPU de 8 bits.
2. Quina és la quantitat màxima que pot representar un registre de 8 bits interns? I de 16 bits?
3. Calcula quants bytes són 3KB, 3KiB, 1.5MB, 1.5 MiB.
4. Si ens venen un disc dur de 1TB, a quants TiB equival?

### 1.3. Operacions bit a bit

Una operació bit a bit (en anglès **bitwise operation**) opera sobre nombres binaris a nivell dels seus bits individuals.

És una acció primitiva ràpida i és suportada directament pels processadors.

En processadors simples de baix cost, les operacions de bit a bit, juntament amb els d’addició i subtracció, són substancialment més ràpides que la multiplicació i la divisió, mentre que en els moderns processadors d’alt rendiment usualment les operacions es realitzen a la mateixa velocitat.

### 

### 1.4. Tipus d’operacions

Es poden distingir fins a tres tipus diferents d’operacions a nivell de bit:

**Operacions bit a bit**

Executen les operacions lògiques AND, OR, XOR, NOT, etc., sobre els bits individuals dels operands.

**Operacions de desplaçament**

Desplacen els bits dels operands cap a la dreta o cap a l’esquerra una o més posicions.

**Operacions de rotació**

Roten els bits de l’operand cap a la dreta o cap a l’esquerra una o més posicions. Es pot, o no, usar el flag de ròssec com un bit addicional en la rotació.

### 

### 

### 

### 1.5. Operacions bit a bit

#### 1.5.1. NOT

El NOT bit a bit, o bitwise, o complement, és una operació unària que realitza la negació lògica a cada bit, invertint els bits del nombre, de tal manera que els zeros es converteixen en 1 i viceversa. Per exemple:

NOT 0111 (decimal 7)

= 1000 (decimal 8)

#### 

#### 1.5.2. AND

El AND bit a bit, o bitwise, presa dos nombres enters i realitza l’operació AND lògica a cada parell corresponent de bits. El resultat en cada posició és 1 si el bit corresponent dels dos operands és 1, i 0 en cas contrari, per exemple:

0101 (decimal 5)

AND 0011 (decimal 3)

= 0001 (decimal 1)

El AND pot ser usat per filtrar determinats bits, permetent que uns bits passin i els altres no.

#### 

#### 1.5.3. OR

Una operació OR de bit a bit, o bitwise, presa dos nombres enters i realitza l’operació OR inclusiu a cada parell corresponent de bits. El resultat en cada posició és 1 si el bit corresponent d’una dels dos operands és 1, i 0 si tots dos bits són 0, per exemple:

0101

OR 0011

= 0111

#### 

#### 1.5.4. XOR

El XOR bit a bit, o bitwise, presa dos nombres enters i realitza l’operació OR exclusiu a cada parell corresponent de bits. El resultat en cada posició és 1 si el parell de bits són diferents i zero si el parell de bits són iguals.

0101

XOR 0011

= 0110

#### 

#### 1.5.5. Operacions de desplaçament

En aquestes operacions els dígits (bits) són moguts, o desplaçats, cap a l’esquerra o cap a la dreta.

Els registres en un processador d’ordinador tenen una amplada fixa, així que alguns bits "seran desplaçats cap a fora" ("shifted out"), és a dir, "surten" del registre per un extrem, mentre que el mateix nombre de bits són "desplaçats cap a dintre" ("shifted in"), és a dir, "entren" per l’altre extrem, les diferències entre els operadors de desplaçament de bits estan en com aquests determinen els valors dels bits que entren al registre (desplaçament cap a dins) (shifted-in).

Per exemple. Si es té en un registre de 8 bits el valor 10110011, i es fa un desplaçament cap a l’esquerra d’un bit, tots els bits es mouen una posició cap a l’esquerra, el bit de l’esquerra es perd i entra un bit zero de farciment pel costat dret. En un desplaçament d’un bit cap a la dreta passa alguna cosa semblant, el bit de la dreta es perd i el de l’esquerra s’omple amb un zero:

|  |  |  |
| --- | --- | --- |
| 10110011 <-- Bits abans del desplaçament | | |
| desplaçament  cap a l'esquerra | desplaçament  cap a la dreta |  |
| <-- 0110011X <-- | --> X1011001 --> | <-- Desplaçament |
| 01100110 | 01011001 | <-- Bits després del desplaçament |

#### Exercicis

1. Calcula amb 4 bits:

3 AND 6

8 OR 1

9 AND 8

NOT 12

13 XOR 15

1. Quin és el resultar de desplaçar 5,8,4,3:

* un bit a l’esquerra?
* dos bits a l’esquerra?

1. Fes el mateix que a l’apartat anterior però amb els desplaçaments a la dreta.

## 2. Estructura d’un computador

Hem de distingir entre dues estructures fonamentals que conformen un ordinador:

L’**estructura física**, formada per tots els components físics que conformen el maquinari de l’ordinador.

L’**estructura funcional**, formada per les agrupacions lògiques d’elements físics o parts d’aquests elements físics que permeten a l’ordinador implementar una funcionalitat concreta, és a dir, realitzar alguna tasca concreta d’entre les múltiples que ha de realitzar el maquinari d’un ordinador.

### 2.1. Estructura física

A grans trets, un ordinador modern consisteix en:

* un (o més d’un) processador
* memòria
* busos
* components de I/O (o perifèrics)

Aquests components s’interconnecten d’alguna manera per aconseguir la funció principal de l’ordinador, és a dir **executar programes**.

#### 

#### 

#### 2.1.1. Memòria Principal

Un ordinador incorpora diversos tipus de memòria, com la memòria principal, la memòria cache, els registres de la CPU, i alguns perifèrics que actuen com a memòria secundària.

**Memòria principal**

La **Memòria principal** està directament connectada a la CPU, és una categoria de memòria que s’utilitza per emmagatzemar informació que se suposa que s’està utilitzant activament.

És l’encarregada d’emmagatzemar les dades i els programes mentre s’estan executant en l’ordinador.

**Memòria RAM**

La memòria d’accés aleatori (**RAM**, Random Access Memory) és aquella en què es triga el mateix temps a accedir a qualsevol cel·la de la memòria. És una memòria volàtil (es perd la seva informació quan s’interromp l’alimentació elèctrica)

De forma interna, la memòria RAM es pot entendre com una taula de cel·les de dades disposades en files i columnes. Per accedir a una dada concreta, continguda en una d’aquestes cel·les, el controlador de memòria ha de donar-li les coordenades on es troba la dada.

**Memòria ROM**

Memòria de només lectura (**ROM**, Read Only Memory) que conté la informació que no es pot modificar, i serveix bàsicament per poder iniciar el sistema informàtic.

|  |  |
| --- | --- |
|  | Actualment gairebé no s’utilitzen les memòries ROM pures, la majoria de memòries de només lectura són, de fet, regrabables.  Les memòries EPROM (*Erasable Programmable Read-Only Memory*) i flash EEPROM es poden esborrar i tornar a programar varies vegades. |

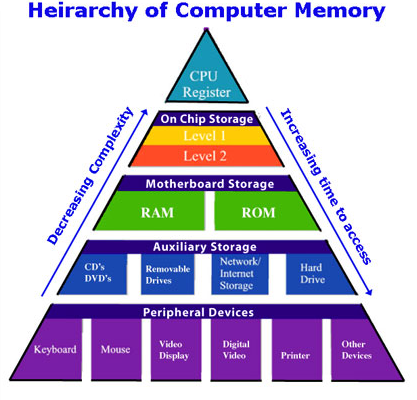
**Memòria caché**

A més a més de la memòria principal, és habitual que els ordinadors incorporin un altre tipus de memòria, anomenada memòria caché, per tal d’agilitzar els càlculs que realitzen els programes. Acostumen a ser memòries **intermèdies entre la RAM i el processador**. Aquestes s’encarreguen d’emmagatzemar temporalment la informació que s’ha de processar i utilitzar amb freqüència. Estan ubicades físicament dins de la CPU i és una de les memòries més ràpides de l’ordinador.

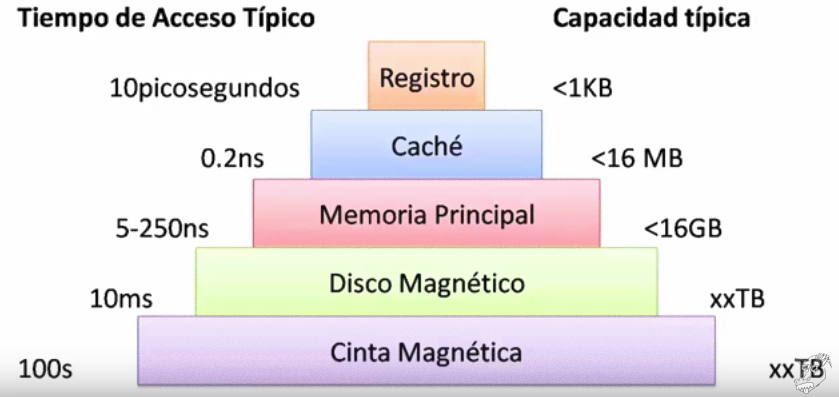
**Memòria externa o secundària**

Reben aquest nom els suports d’emmagatzemament massiu, ja que poden guardar gran quantitat d’informació de forma permanent. Per exemple: Disc dur, CD, DVD’s…​

Aquest tipus de memòria és més lenta que la memòria principal, ja que està formada per elements electrònics i mecànics. És memòria no volàtil, això vol dir que la informació roman en ella, fins i tot després d’interrompre el subministrament elèctric.



*Figura 8. Jerarquia de les memòries*



*Figura 9. Velocitat i capacitat de les diferents memòries*

Quant parlem de memòries són importants els termes següents:

**Cel·la de Memòria**

Dispositiu o circuit elèctric que s’usa per emmagatzemar un sol bit (0 o 1). **Quan parlem de la capacitat d’una memòria estem fent el recompte de les cel·les que conté**.

Per exemple, una memòria de **4GB** té **4·1024·1024·1024·8 bits o cel·les**

**Paraula de Memòria**

Representa la quantitat de bits que pot manipular un ordinador a la vegada. La mida d’una paraula depèn de l’arquitectura de l’ordinador. La mida de la paraula és generalment igual al nombre de bits utilitzats per representar un nombre enter i la longitud d’una instrucció dins de la CPU. Els ordinadors moderns normalment tenen una mida de paraula de 32 o 64 bits.

**Word-addressable**

Si l’arquitectura d’un ordinador és word-addressable vol dir que el processador **només pot adreçar paraules completes a la memòria**. És a dir, les adreces de memòria apunten a paraules, no a bytes concrets.

Algunes arquitectures són **byte-addressable**.

|  |  |
| --- | --- |
|  | La longitud de la paraula no té perquè coincidir amb l’espai d’adreçament. |

#### 

#### Exercicis

1. Quin tipus de memòria és més ràpid? Quina és la més lenta?

Els registres son les més ràpides i les més lentes son les cintes magnètiques.

1. A igualtat de preu, quina memòria és més gran? Quina és més petita?

El disc magnètic es més gran i el més petit la memòria principal.

1. Quina diferència hi ha entre la ROM i la EEPROM?

La ROM no es pot modificar res i l’altre sí que es pot.

1. Troba informació de la mida de la caché de la CPU i7-3770 de Intel.

#### 

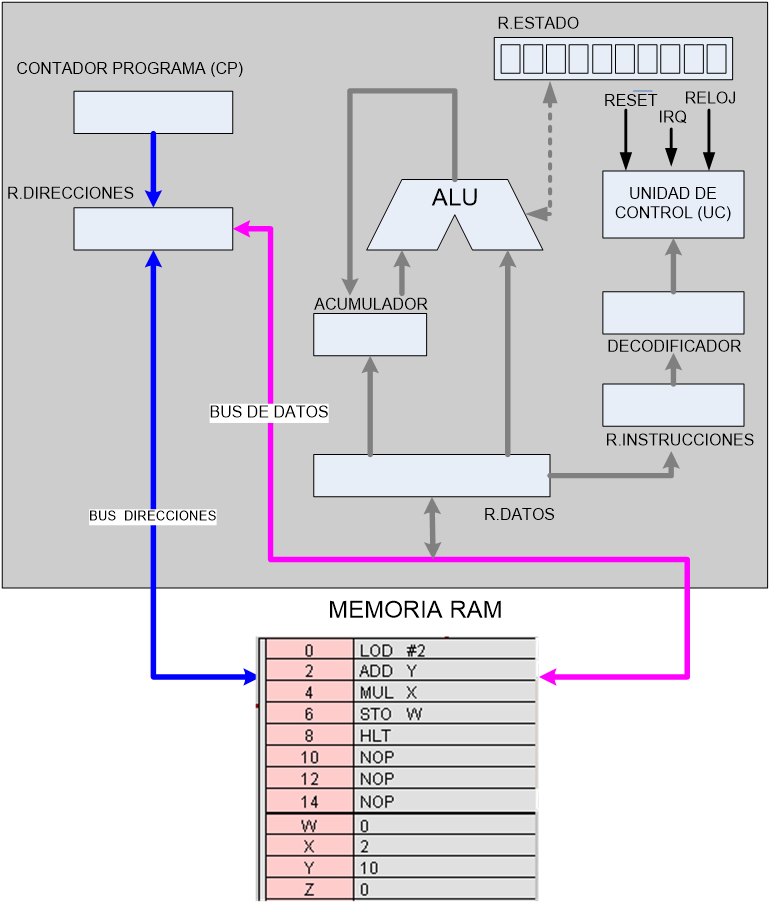
#### 

#### 2.1.2. La CPU ( Central Processing Unit)

Arquitectura de Von Neumann

L’arquitectura de Von Newmann és una arquitectura de computadors basada en la descrita pel matemàtic i físic John Von Neumann el 1945.

L’arquitectura descriu un model conceptual de com funciona un ordinador En aquest model la CPU bàsicament està formada per l'**ALU** (unitat aritmètico-lògica), la **CU** (unitat de control), un conjunt de memòries molt ràpides (i molt petites) que anomenem **registres** interns i un **rellotge** per sincronitzar-ho tot.



*Figura 2. Model de Von Neumann*

*Hello World en codi màquina amb comentaris*

b8 21 0a 00 00 #moving "!\n" into eax

a3 0c 10 00 06 #moving eax into first memory location

b8 6f 72 6c 64 #moving "orld" into eax

a3 08 10 00 06 #moving eax into next memory location

b8 6f 2c 20 57 #moving "o, W" into eax

a3 04 10 00 06 #moving eax into next memory location

b8 48 65 6c 6c #moving "Hell" into eax

a3 00 10 00 06 #moving eax into next memory location

b9 00 10 00 06 #moving pointer to start of memory location into ecx

ba 10 00 00 00 #moving string size into edx

bb 01 00 00 00 #moving "stdout" number to ebx

b8 04 00 00 00 #moving "print out" syscall number to eax

cd 80 #calling the linux kernel to execute our print to stdout

b8 01 00 00 00 #moving "sys\_exit" call number to eax

cd 80 #executing it via linux sys\_call

L’executable es passa a memòria principal i la CPU el recupera, instrucció per instrucció, i el va executant. Per poder executar un programa és imprescindible que aquest estigui carregat a la memòria principal.

La CPU bàsicament està formada per l'**ALU** (unitat aritmètico-lògica), la **CU** (unitat de control), un conjunt de memòries molt ràpides (i molt petites) que anomenem **registres** interns i un **rellotge** per sincronitzar-ho tot.

**ALU**

Composta per un circuit combinacional complex que s’encarrega de realitzar les operacions aritmètiques (suma, multiplicació…) i lògiques (AND, OR …) i retorna el resultat, tot sota la supervisió de la CU. Dins de l’ALU es distingeixen dues parts importants:

**Acumulador**

És un registre d’emmagatzematge temporal que servirà per a emmagatzemar operands i resultats intermedis de les operacions. En els ordinadors actuals existeixen una colla de registres de propòsit general que es poden utilitzar per aquesta funció, però que no són exclusivament acumuladors.

**Registre d’estat**

És un registre on s’emmagatzema informació sobre el resultat que s’obté a la darrera instrucció (si el resultat és 0, si és parell o si és tan gran que no s’ha pogut emmagatzemar etc.)

**UC**

És la part encarregada d’interpretar les instruccions i d’enviar les ordres precises a tots els elements per al correcte processament de les dades i els monitoritza. Dins de l’UC trobem dos elements molt importants:

**Descodificador d’instruccions (DI)**

Interpreta les instruccions del programa i les tradueix a micro-ordres gravades al xip que desencadenaran accions en diferents components.

**Registres interns**

Cada registre pot emmagatzemar **una dada** (tants bits com paraula té el processador). La quantitat de registres que té una CPU depèn de la seva arquitectura concreta, però n’hi ha alguns que són bàsics per entendre el seu funcionament:

**Comptador de programa (PC)**

Manté, en tot moment, l’adreça de memòria de la següent instrucció que s’executarà.

**Registre d’instrucció (IR)**

En aquest registre es guarden les instruccions moments abans de descodificar-les en el DI.

**Registre d’adreces (MAR)**

Emmagatzema l’adreça actual on s’accedirà. Es bolca al bus d’adreces i al arribar a la memòria, el que contingui aquesta adreça es bolca al bus de dades.

**Registres de dades (MDR)**

Registres on s’emmagatzemen les dades de l’operació actual. Aquestes dades poden provenir de dos llocs, de la memòria principal via bus de dades o de l’acumulador com a resultat d’una operació via registre de dades passant pel bus de dades i d’allà a la memòria.

**Punter pila (SP)**

Una pila és una estructura dinàmica (LIFO Last In First Out) en la qual s’afegeixen i eliminen elements només per un extrem, anomenat capçalera de la pila. Sobre una pila es poden realitzar dues operacions: afegir (push) un element i treure (pop) un element. És la tècnica que es segueix per emmagatzemar les instruccions quan es fan crides a subrutines. El punter pila guarda la direcció de la capçalera de la pila de subrutines.

**Rellotge**

Cristall de quars que genera unes ones quadrades de freqüència constant i que serveixen per a que tot el sistema vagi sincronitzat. La freqüència d’aquest rellotge es mesura en MHz o GHz i determina la velocitat del processador.

#### Exercicis

1. A l’exemple de codi màquina, sabries trobar en quin format estan les dades (b8 21 0a …)?
2. A la figura 2 a la taula de sota on diu RAM, digues què representen les dues columnes. Digues també perquè a la taula hi ha una separació entre les primeres files i les últimes quatre.
3. Què són les operacions lògiques AND, OR i NOT?
4. Digues en cada cas, en quin registre s’emmagatzemen a la CPU:

* l’adreça actual de la RAM on es troba la instrucció que s’execurà a continuació
* la instrucció actual que s’executarà abans de ser decodificada
* l’adreça a la qual es vol accedir i que té una dada que es vol recuperar
* les dades
* la pila

1. En el llistat de codi màquina del *Hello World* amb comentaris, què creus que són *eax, ebx, ecx* i *edx*?
2. A què ens referim quan parlem d’una estructura LIFO? Quina funció fa la pila?
3. Troba quina forma té el voltatge del senyal de rellotge (o clock).

#### 

#### 2.1.3. Busos de comunicació

La placa base és un element fonamental d’un ordinador. No només és el suport físic de tots els altres elements, sinó que proporciona els circuits necessaris per comunicar-los.

**Un bus és un subsistema que transfereix dades entre components d’ordinador dins d’un ordinador o entre ordinadors.**

Dins d’un ordinador tots els components estan units utilitzant diferents busos, de manera que les dades puguin viatjar d’un component a qualsevol altre. Cada bus consta d’un camí que permet comunicar selectivament un cert número de components o dispositius d’acord amb unes determinades normes de connexió.

Les pistes d’un bus que recorre la placa base segons la seva utilització en:

**Bus de dades**

Transporten les dades d’un dispositiu a un altre, normalment de la CPU a un dispositiu o viceversa (exceptuant DMA). És per tant un bus bidireccional.

**Bus d’adreces**

Identifica el dispositiu i la posició a la que va destinada la informació que es transmet pel bus de dades. És un bus unidireccional.

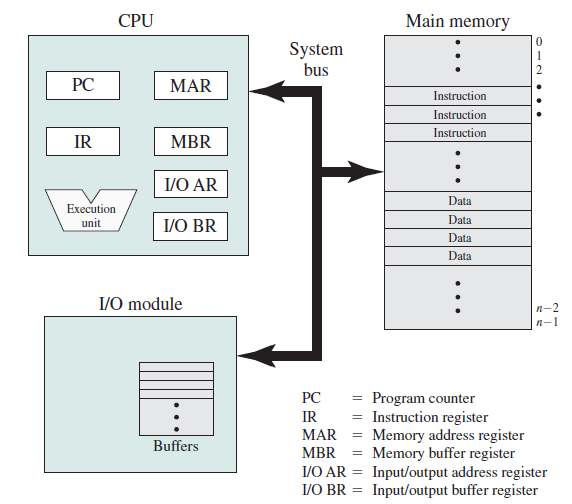
**Bus de control**

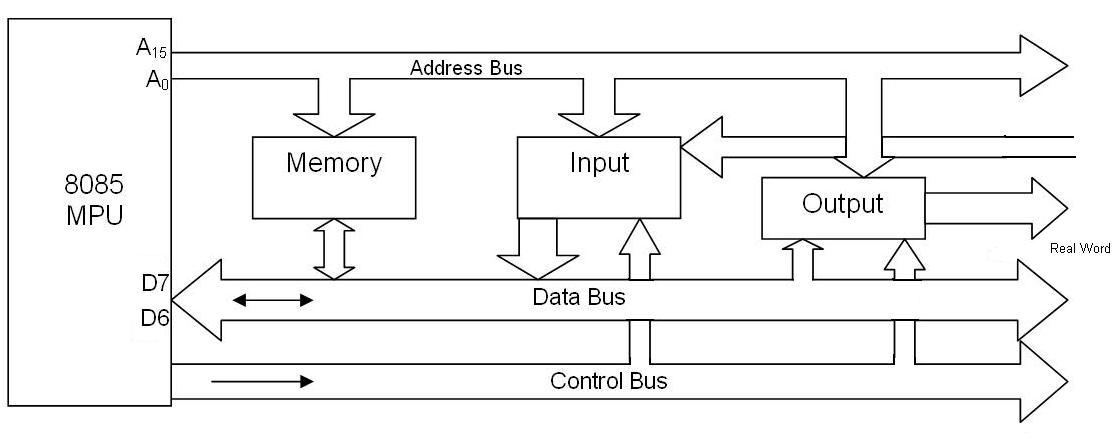
Transporta informació útil per gestionar el sistema. **Organitza i redirigeix la informació cap al bus que correspon segons la informació que s’ha de transmetre**. S’encarrega de fer l’adreçament de memòria. És un bus bidireccional.

|  |  |
| --- | --- |
|  | Existeix una estreta relació entre aquests tipus de busos, ja que per a cada dada o instrucció enviada per un dels busos, existeix una adreça que es transporta per l’altre bus que indica l’origen o destí d’aquesta dada o instrucció. |

La velocitat (o freqüència) del bus queda determinada pels impulsos del rellotge. A més impulsos, més MHz i per tant més velocitat el bus.

El tipus de bus i la seva velocitat depenen del fabricant i del processador. Podem ampliar memòria, canviar disc dur i processador, però si no canviem el bus, la CPU serà més ràpida, però la transferència de dades serà la mateixa. Això es coneix com a coll d’ampolla.





#### 

#### Exercicis

1. Suposa un sistema amb un bus d’adreces de 8 bits, un bus de dades de 4 bits i un bus de control d’un bit (R/W). Com quedarien les línies dels busos si:
   1. La CPU vol escriure un 10 a la posició 67.
   2. La CPU vol llegir un valor de la posició 135 de la RAM
2. La tarja gràfica també es comunica mitjançant busos amb la CPU i la RAM? En cas afirmatiu, troba quina mena de busos fa servir.

#### 

#### 

#### 2.1.4. Les unitats d’entrada/sortida

El concepte d’entrada/sortida fa referència a qualsevol comunicació o intercanvi d’informació entre la CPU mitjançant la memòria principal amb l’exterior. Les entrades són les senyals que rep la unitat, i les sortides són les senyals que envia.

Hi ha **dispositius d’entrada**, com els teclats o el ratolins, i **dispositius de sortida**, com els monitors o les impressores.

Altres **dispositius són d’entrada/sortida** com els DVD, disc dur o les NICs (Network Interface Card, RJ45).

Per tal que es pugui dur a terme aquest intercanvi d’informació, aquests elements funcionals han de ser capaços de dur a terme les següents tasques.

**Adreçament**

selecció del dispositiu d’entrada/sortida implicat a cada transferència concreta.

**Transferència**

intercanvi d’informació cap a o des del dispositiu.

**Sincronització**

procés d’acord i supervisió del procés de transferència de dades entre la CPU i el perifèric.

El SO ha d’oferir a la resta del sistema una interfície standard, simple i uniforme per a l’ús del dispositiu.

Aquestes unitats oculten els detalls de implementació dels dispositius externs, de forma que la CPU pugui comunicar-se amb ells amb simples ordres de lectura i escriptura, com si es tractés d’un espai de memòria.

Per tal de fer arribar informació als dispositius i que aquests puguin enviar informació a la CPU s’utilitza el sistema d’assignació de ports d’E/S. Un port d’E/S és una petita zona de memòria (64 bytes com a molt) que s’assigna al dispositiu, de manera que tot l’intercanvi d’informació entre el dispositiu i la CPU es realitza a través d’aquestes adreces.

|  |  |
| --- | --- |
|  | No hem de confondre aquests ports d’E/S, que són adreces de memòria, amb els connectors externs de la placa base, que també s’anomenen sovint ports d’entrada/sortida. |

#### 

#### Exercicis

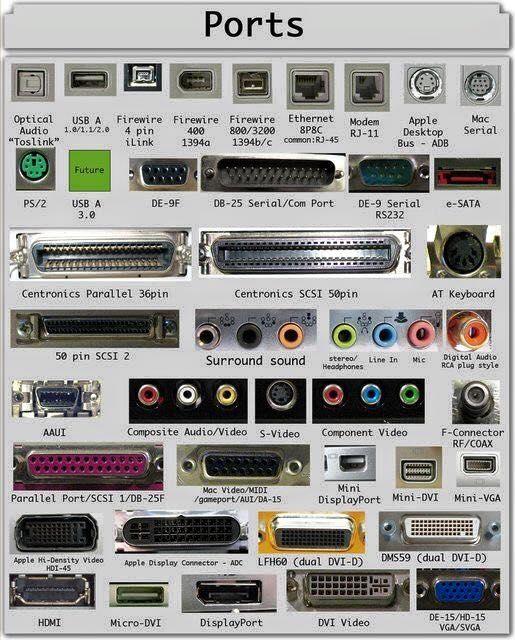
1. Digues quins d’aquests es consideren dispositius d’E/S: CPU, disc dur, impressora, RAM, memòria caché, tarja de so, tarja de xarxa integrada amb la placa.

#### 

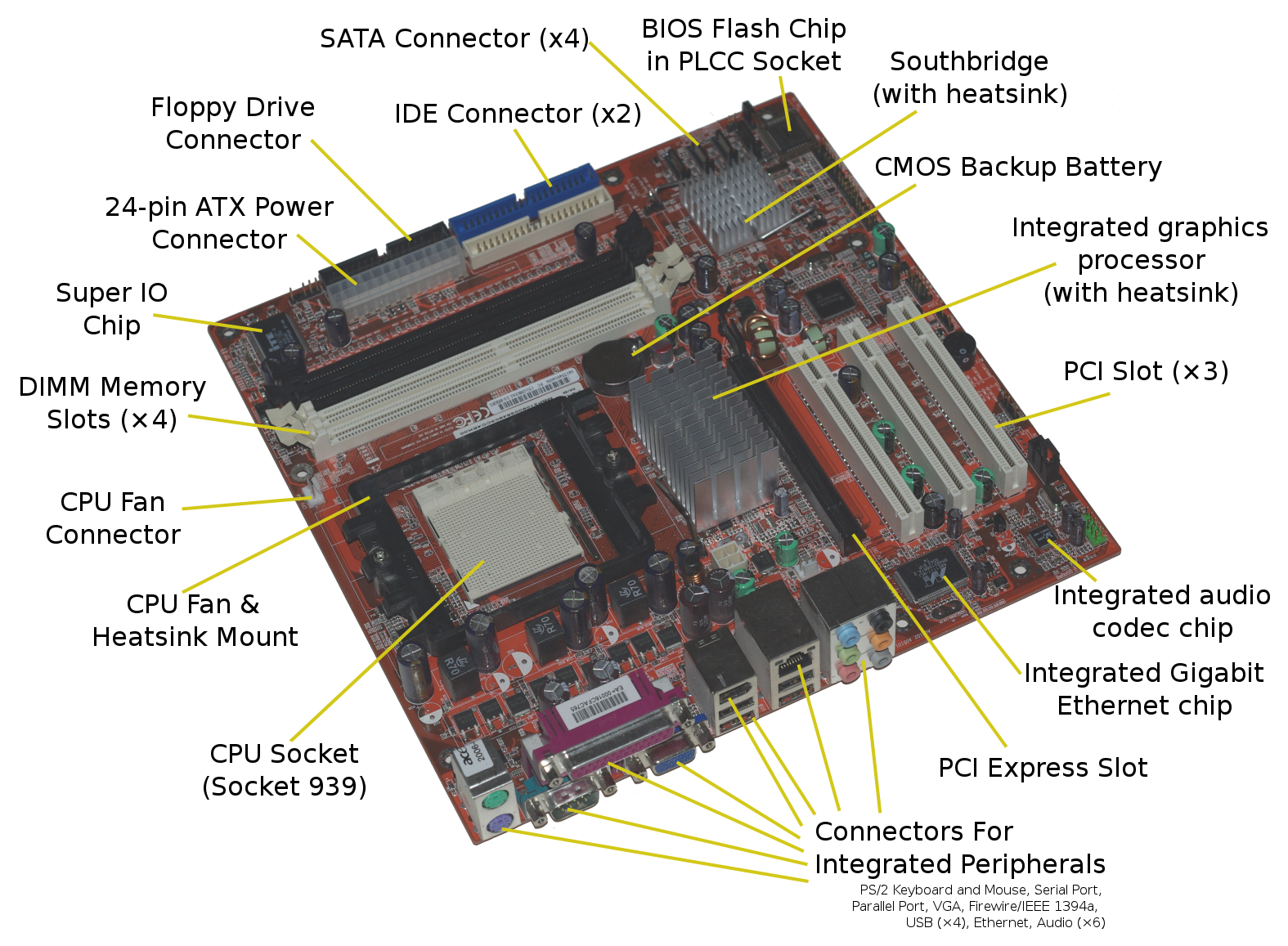
#### 2.1.5. Ports

Els perifèrics es connecten a l’ordinador a través dels ports. Aquesta gestió la fa la unitat d’entrada/sortida.

Aquests ports són connectors que permeten que les dades entrin/surtin de l’ordinador. Els connectors es fan servir per a unir els dispositius perifèrics a l’ordinador. El connector pot estar situat en l’extrem de cable del dispositiu perifèric o unit a aquest i es fica dins d’un port per tal de fer la connexió entre l’ordinador i el perifèric.



*Figura 12. Diferents tipus de ports*



*Figura 13. Parts d’una placa mare*

#### Exercicis

1. Al teu ordinador, identifica els ports: USB, vídeo (HDMI, VGA, DVI, Display Port), audio (micro, auriculars, line in) i Ethernet.
2. A la següent placa mare:

<https://cdn.thinglink.me/api/image/629249298864799744/1240/10/scaletowidth>

Identifica: El socket de la CPU, els bancs de memòria, els connectors PCI, els connectors PCI Express i els connectors SATA per al disc dur.

1. Troba informació sobre la diferència entre l’arquitectura CISC i RISC.



Referències:

<https://debugger.medium.com/why-is-apples-m1-chip-so-fast-3262b158cba2>

<https://electronics-club.com/risc-and-cisc-architecture-characteristics-and-advantages/>

<https://www.microsiervos.com/archivo/ordenadores/nandgame-juego-logica-binaria-circuitos.html>

[https://github.com/myTeachingURJC/Arq-computadores-01/wiki/S05#el-computador-nanorisc-v](https://github.com/myTeachingURJC/Arq-computadores-01/wiki/S05" \l "el-computador-nanorisc-v)