

## UD 03\_01 – Introducció al Hardware.

# Funcionament i Arquitectura de l'Ordinador.

### CONTENIDO

<b>1 Introducció.....</b>	<b>2</b>
<b>2 Components físics.....</b>	<b>2</b>
2.1. Unitat Central de Processament.....	3
2.1.1. Unitat de Control.....	3
2.2. . La Unitat Aritmeticològica.....	4
2.3. Memòria.....	5
2.3.1. Manera de funcionament.....	6
2.4. La unitat d'entrada/eixida i els perifèrics.....	8
2.5. Busos de comunicació.....	10
<b>3 Sistemes lògics.....</b>	<b>11</b>
3.1. Format de les instruccions.....	11
3.2. Maneres d'adreçament.....	13
3.2.1. Adreçament immediat.....	13
3.2.2. Adreçament Directe.....	14
3.2.3. Adreçament indirecte.....	14
3.2.4. Adreçament relatiu.....	15
3.3. Fases d'una instrucció.....	15
3.3.1. Fase de càrrega d'una instrucció.....	15
3.3.2. Execució d'una instrucció.....	16

# 1 INTRODUCCIÓ

Les primeres màquines de computació treballaven amb programes fixos. Alguns ordinadors molt simples encara utilitzen aquest disseny, per a propòsits de simplicitat o entrenament. Per exemple, una calculadora (en principi) és un ordinador de programa fix. Pot fer matemàtiques bàsiques, però no es pot utilitzar com a processador de textos o fer córrer videojocs. Per canviar el programa de tal màquina, ha de reconnectar, reestructurar, o fins i tot redissenyar-la. Podem dir doncs que els primers ordinadors no estaven "programats" sinó "dissenyats". Reprogramar-los sovint significava tornar a començar amb el disseny per finalment aplicar-lo en forma de canvis físics, modificant centenars de connexions.

La idea de l'ordinador de programa emmagatzemat deixava enrere tots aquests problemes. Creant una arquitectura amb un joc d'instruccions prou ample i detallant el càlcul com a sèrie d'instruccions (el que anomenem "programa"), la màquina es torna molt més flexible. Tractant aquelles instruccions de la mateixa manera com dades, una màquina de programa emmagatzemat pot fàcilment canviar el programa utilitzat, fins i tot tenir-ne diversos "carregats" de manera que el canvi sigui instantani.

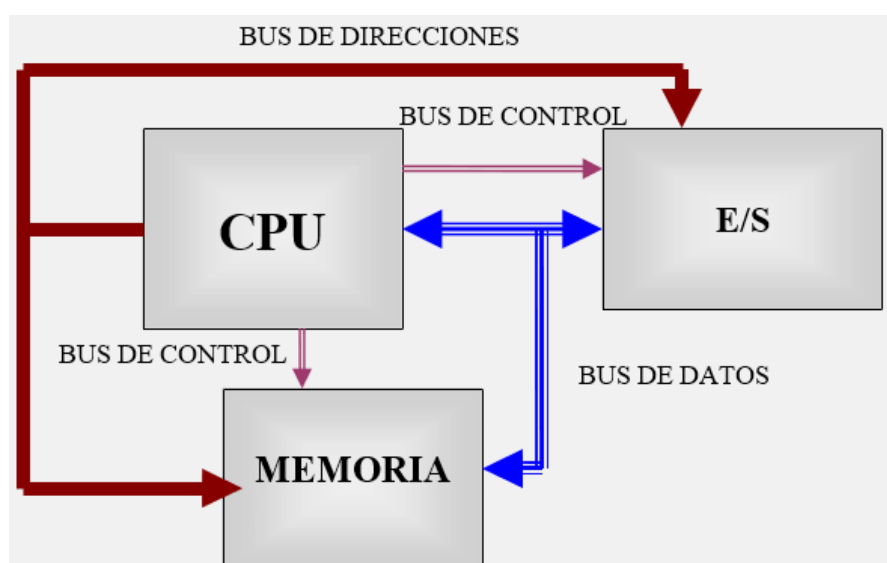
Els termes "arquitectura de von Neumann" i "ordinador de programa emmagatzemat" són generalment utilitzats de manera intercanviable. En aquest tema, descriurem els components físics d'un sistema informàtic a partir de l'arquitectura von Neumann.

## 2 COMPONENTS FÍSICS

Els principals components físics d'un sistema informàtic són:

- la unitat central de processament, CPU.
- la memòria,
- la unitat d'entrada/eixida
- i uns elements que serveixen d'unió a tots els anteriors anomenats busos.

Tots ells es relacionen entre si segons el següent esquema conegut com a arquitectura von Neumann:



## 2.1. UNITAT CENTRAL DE PROCESSAMENT

Aquesta unitat sol abreujar-se com UCP (Unitat Central de Processament), però és molt freqüent l'ús de l'abreviatura anglesa CPU (Central Processing Unit).

Físicament està format per circuits de naturalesa electrònica que es troben integrats en una pastilla o xip denominada microprocessador.

La unitat central de processament és un element molt important en una computadora i s'encarrega d'organitzar tot el treball i realitzar els càlculs aritmètics i lògics; així com de supervisar totes les operacions del sistema informàtic. Sovint es diu d'ella que és com el cervell que controla i governa tot el sistema.

Com aconsegueix fer tot això?

Per a això extrau, una a una, les instruccions del programa que es té allotjat en un altre dels components físics denominat memòria central, les analitza i emet les ordres necessàries per a la seua completa realització.

La unitat central està connectada amb la memòria i amb els perifèrics per mitjà d'uns canals, a través dels quals circula la informació. Aquests canals reben el nom de busos.

La informació que circula entre la unitat central i la memòria és de dos tipus: les **instruccions** i les **dades**.

- Les instruccions són les ordres que configuren el programa que la computadora ha d'executar.
- Les dades són conjunts de símbols que representen un valor.

La unitat central està formada per dues parts clarament diferenciades:

- la unitat de control
- i la unitat aritmeticològica.

### 2.1.1. UNITAT DE CONTROL

La unitat de control s'encarrega de governar la resta d'unitats. Per a governar necessita enviar missatges i enviar les dades a processar per uns canals especials que també governa. Totes aquestes ordres anomenades senyals de control i les dades van canalitzats per unes vies anomenades busos.

Per a fer la seua tasca, la UC necessita conèixer, d'una banda, la instrucció i, per una altra, una sèrie d'informacions addicionals que haurà de tindre en compte per a coordinar, de manera correcta, l'execució de la instrucció. El resultat de la interpretació d'aquestes informacions són una sèrie d'ordres als diferents elements de la computadora.

Com a informacions addicionals a les instruccions podem veure els impulsos de rellotge i els indicadors d'estat. Els impulsos de rellotge sincronitzaran totes les activitats que es duguen a terme. Els indicadors d'estat són una sèrie de valors que es modifiquen segons resultats de les operacions anteriors guardant una memòria històrica dels esdeveniments precedents perquè, en funció d'aquests esdeveniments, pugui la UC prendre decisions.

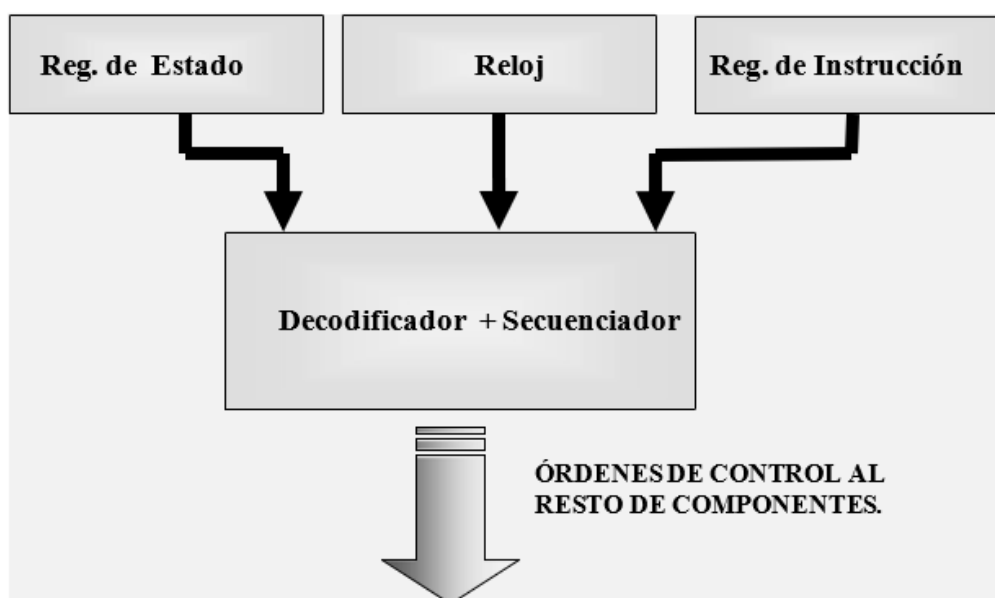
Els **components** principals de la UC són:

- Un element format per dos components que descodifica i seqüència les instruccions. El primer element descodifica la instrucció per a veure de què es tracta, això ho realitza un **descodificador**. I

una vegada sap de què es tracta, el segon anirà executant la instrucció pas a pas, és a dir una instrucció no s'executa en un sol pas, sinó que és una seqüència de passos, necessitant cada pas per a la seua realització els senyals de control corresponents. Així doncs el **seqüenciador** és l'encarregat de generar en cada pas els senyals de control necessàries.

- Diversos elements de memòria denominats **registres**:
  - El registre que emmagatzema la instrucció mentre l'interpret està traduint el seu significat: Registre d'Instrucció (RI). La resta de les instruccions romanen en la memòria, esperant que els toque el seu torn d'execució.
  - El registre que guarda quina és la direcció de la pròxima instrucció, és a dir, guarda la referència de la següent instrucció que s'ha de realitzar, per a poder anar a buscar-la una vegada que finalitza l'execució de la instrucció en curs; aquest registre es coneix amb el nom Comptador de Programa (CP).
  - Els indicadors d'estat estan agrupats en un registre denominat Registre d'Estat (RE).

En la següent figura s'esquematitza el conjunt de senyals que utilitza la UC i les que genera.



## 2.2. . LA UNITAT ARITMETICOLÒGICA

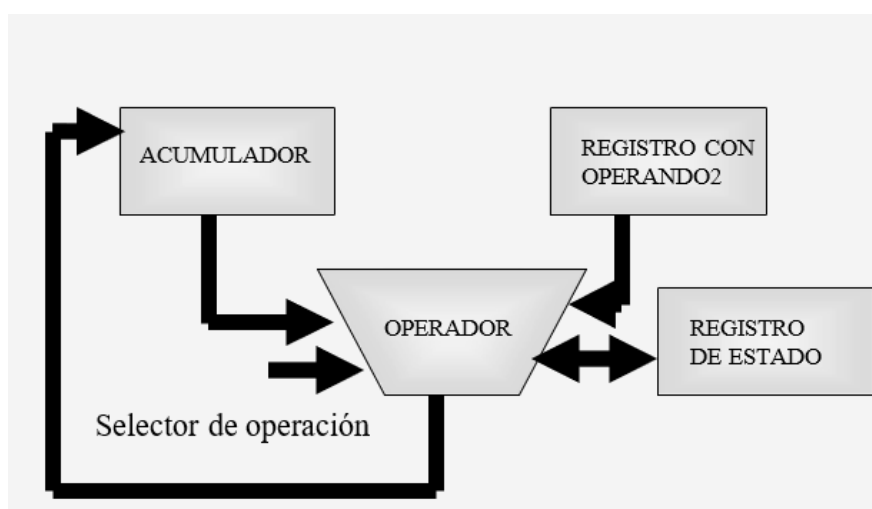
La Unitat Aritmeticològica (UAL) és l'encarregada de realitzar els càlculs; sempre sota la supervisió de la unitat de control. Les dades sobre els quals es realitzen les operacions es denominen operands. A l'element encarregat d'executar les operacions se'l denomina operador, i aquesta format per una sèrie de circuits electrònics que són capaços de sumar dos nombres binaris o fer les operacions lògiques elementals: disjunció, conjunció i negació.

Perquè l'operador realitzi l'operació, els operands s'emporten a la UAL i es guarden en uns registres denominats **registres de treball**. El resultat de l'operació es guarda també en un registre abans de ser portat a la memòria o a la Unitat d'Entrades i Eixides. Sovint s'utilitza un mateix registre per a guardar un dels operands i, també, el resultat, a aquest registre se'l coneix amb el nom d'**acumulador**.

L'operador, a més de calcular el valor de l'operació, modifica el registre d'estat segons el resultat de l'operació. Així, si el resultat és un valor negatiu, es modifica un bit (quantitat mínima d'informació en binari. El binari és el llenguatge intern dels ordinadors. En binari només hi ha dos valors possibles: 0 i 1) d'aquest registre, anomenat bit negatiu o bit N, posant-se a 1; per contra, el bit N romandrà en estat 0 mentre el contingut de l'acumulador no siga negatiu. D'igual forma indicara la UAL a la UC si el resultat ha sigut zero, o si ha produït algun ròssec<sup>1</sup>, etc.



Unitat aritmètica sense acumulador



Unitat aritmètica amb acumulador

## 2.3. MEMÒRIA

La memòria és la unitat on s'emmagatzemen les dades i les instruccions que constitueixen els programes.

Hi ha dos tipus bàsics de memòria: memòria interna o central i memòria externa o massiva. En aquest apartat ens ocuparem de la memòria central o també coneguda com a memòria RAM o principal i deixarem per a més endavant la memòria externa (discos, cintes ...).

La memòria no és res més que un conjunt ordenat de cel·les o posicions de memòria, numerades de manera consecutiva, capaces de retindre informació, mentre la computadora està connectada. Per a

<sup>1</sup> En castellà, *acarreo*, en anglès, *carry*.

entendre-ho imagineu un enorme caseller on en cada casella poguérem emmagatzemar una combinació de 8 uns i zeros.

A cada posició es pot accedir per mitjà d'un número que la identifica. Aquest número es coneix amb el nom de direcció de memòria. Mitjançant aquesta direcció es pot accedir de manera directa a qualsevol d'elles independentment de la seua posició; es diu, per això, que la memòria central és un suport d'informació d'accés directe.

Direcció de Memòria	Contingut de cada cel·la
0	10101010
1	11010100
2	11010101
....	01010100
1023	00010100

Cal no confondre els termes cel·la o posició de memòria amb el de paraula, ja que aquesta última és la quantitat d'informació que pot introduir-se o extraure's de la memòria central d'una sola vegada. Informació que es pot llegir o escriure en un únic colp de rellotge. La grandària habitual de la paraula sol ser 16, 32, 64 bits.

La grandària de paraula és un conjunt de posicions de memòria.

La grandària de la memòria es mesura per bytes (8 bits) i els seus múltiples:

Bit: unitat mínima de memòria. Es correspon amb un "1" o amb un "0".

$$1\text{B} = 1 \text{ Byte} = 8 \text{ bits}$$

$$1 \text{ Kibibyte} = 1 \text{ KiB} = 2^{10} \text{ bytes} = 1024 \text{ bytes}$$

$$1 \text{ Mebibyte} = 1 \text{ MiB} = 2^{10} \text{ KiB} = 2^{20} \text{ Bytes}$$

$$1 \text{ Gibibyte} = 1 \text{ GiB} = 2^{10} \text{ MiB} = 2^{20} \text{ KiB} = 2^{30} \text{ Bytes}$$

$$1 \text{ Tebibyte} = 2^{10} \text{ GiB} = 2^{20} \text{ MiB} = 2^{30} \text{ KiB} = 2^{40} \text{ Bytes}$$

El kibibyte està relacionat amb el kilobyte, aquest últim indica  $10^3$  bytes = 1000 bytes, encara que de vegades s'utilitzen com a sinònims. Aquest nou prefix de byte es fa servir per evitar aquesta confusió. El mateix es pot dir per a la resta de múltiples.

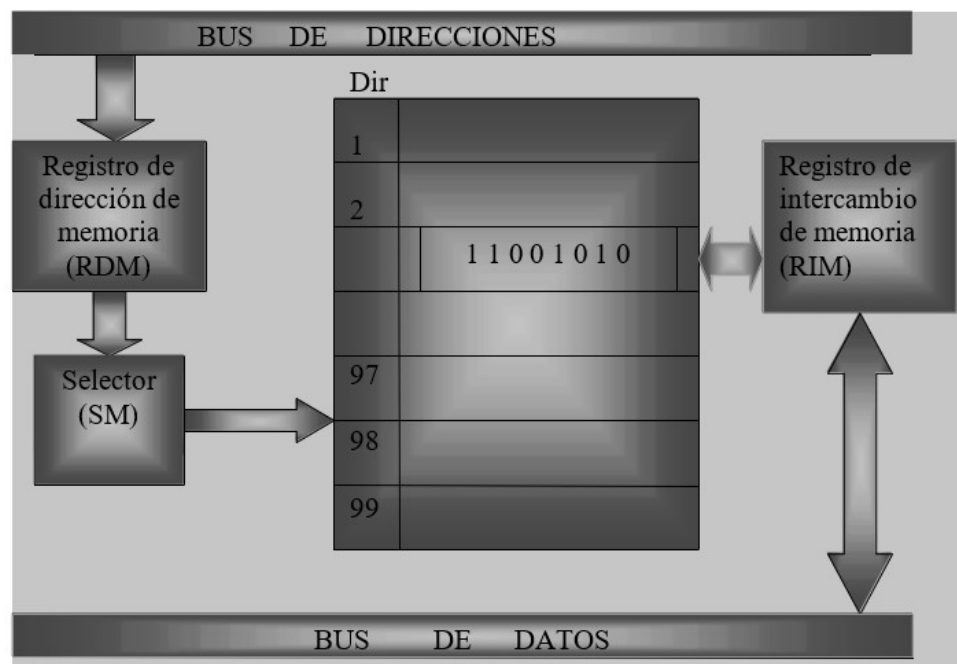
### 2.3.1. MANERA DE FUNCIONAMENT

L'acció de guardar una informació en una paraula de la memòria es diu accés d'escriptura, i l'acció de recuperar-la, accés de lectura. Els accessos són coordinats sempre per la UC.

La memòria té associats dos registres per a la realització de les operacions de lectura / escriptura i un dispositiu encarregat de seleccionar la cel·la de la memòria implicada en una determinada operació.

- Registre de Direcció de Memòria (RDM), abans d'una operació de lectura o escriptura s'ha de col·locar en aquest registre la direcció de la cel·la implicada en l'operació de lectura o escriptura.

- **Registre d'Intercanvi de Memòria (RIM)** : conté la dada a llegir o a escriure.
  - si es tracta d'una operació de lectura de memòria, aquest registre és el que rep la dada de la posició de memòria assenyalada pel RDM.
  - si es tracta d'una operació d'escriptura, aquest registre conté la dada que gravarem en la posició de memòria referida pel RDM.
- **Selector de Memòria (SM)**, aquest dispositiu s'activa cada vegada que es produeix una ordre de lectura o escriptura en memòria, connectant la cel·la de memòria, la direcció de la qual figura en el RDM amb el RIM, possibilitant l'intercanvi d'informació en tots dos sentits.



#### Descripció d'una operació **d'escriptura**.

1. La UC indica a la MP que desitja escriure.
2. La UC envia l'adreça en la qual es posarà la dada. Aquest es guarda el RDM.
3. La UC envia els senyals de control necessàries perquè la dada a escriure s'emmagatzemi en el RIM.
4. La UC dona l'ordre d'inici de l'accés. Després de la qual cosa l'SM connecta la cel·la de memòria indicada pel RDM amb el RIM possibilitant el pas d'informació des del RIM a la cel·la de memòria.
5. La UC espera el temps necessari perquè la MP realitzi l'escriptura.

#### Descripció d'una operació de lectura.

1. La UC indica a la MP que desitja llegir.
2. La UC envia l'adreça de la qual volem llegir. Aquesta es guarda el RDM.
3. La UC dona l'ordre d'inici de l'accés. Després de la qual cosa l'SM connecta la cel·la de memòria indicada pel RDM amb el RIM possibilitant el pas d'informació des de la cel·la de memòria al RIM.

4. La UC espera el temps necessari perquè la MP realitzi l'operació de lectura. Després de la qual queda la dada emmagatzemada en el RIM i d'ací es dirigirà al seu destí seguint de nou les ordres de la UC.

La velocitat de la memòria ve determinada pel temps de cicle i el temps d'accés.

- Temps d'accés de lectura: temps transcorregut des de l'ordre de L sobre la memòria fins que la dada està disponible en RIM.
- Temps d'accés d'escriptura: Temps transcorregut des de l'ordre de E fins que la dada es troba estable en la posició de memòria.
- Temps d'accés: Temps mitjà entre temps d'accés en lectura i temps d'accés en escriptura.
- Temps de cicle: o cicle base de memòria. Temps que necessita la memòria fins que està disposada per a efectuar una altra operació de I/O. Sempre serà major que el temps d'accés.

Normalment aquests temps són fixos per a una memòria i estan condicionats per la grandària de la paraula.

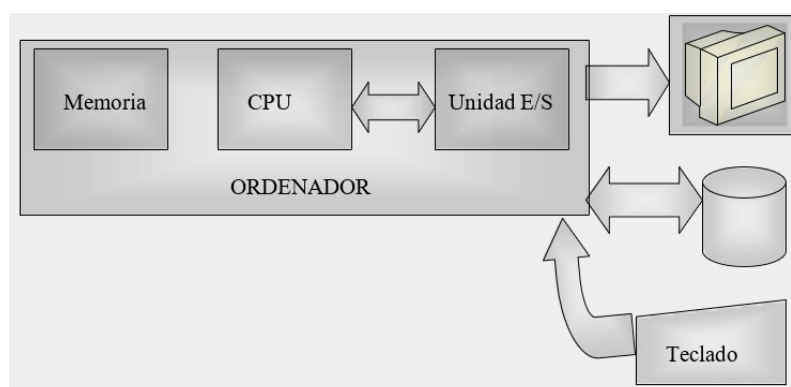
## 2.4. LA UNITAT D'ENTRADA/EIXIDA I ELS PERIFÈRICS

Els perifèrics relacionen a la unitat central i la seua memòria interna amb el món exterior. Hi ha una gran varietat de perifèrics, alguns tan coneguts i necessaris com el teclat, la pantalla o la impressora, i altres més especialitzats, com l'escàner d'imatge òptic. Els perifèrics es divideixen en perifèrics d'entrada: elements que permeten l'entrada de dades des de l'exterior al sistema, per exemple: ratolí, teclat, escàner.... i perifèrics d'eixida que permeten l'eixida d'informació des del sistema a l'usuari; per exemple: pantalla, impressora, altaveus. També existeixen dispositius d'entrada/eixida, que permeten el flux d'informació en els dos sentits.

Entre ells i la CPU es troba la unitat d'entrada/eixida (altres noms: mòdul I/O, controlador).

El perquè de no connectar directament els dispositius externs al bus que comunica la CPU i la memòria radica en aspectes de control, velocitats diferents, grandària de dades diferents... tot això desemboca en una necessitat d'adaptar unes capacitats (de perifèrics) amb altres (CPU). Aquest mòdul I/O té les següents característiques:

- Permet la comunicació entre CPU-Perifèrics.
- Són genèrics: un mòdul de I/O pot gestionar dispositius externs de diversos fabricants
- A vegades poden ser fins i tot autèntics processadors.





La coordinació de la comunicació entre els perifèrics i la CPU la realitza la Unitat de I/O. Observe's que aquesta no és un perifèric sinó un dispositiu que gestiona als perifèrics seguint les ordenes de la CPU; és a dir, la Unitat de I/O rep de la Unitat de Control informació sobre la mena de transferència de dades que ha de realitzar (si és d'entrada o d'eixida) i que perifèric ha d'utilitzar; si és d'eixida rebrà també la dada que ha d'enviar i el moment de l'operació.

Llavors, la Unitat de I/O seleccionarà el perifèric i executarà l'operació tenint en quanta les característiques pròpies de cada perifèric. Una vegada executada l'ordre avisarà a la UC de la terminació de la transferència.

Cada perifèric tindrà assignat un número o direcció que servirà per a identificar-lo. Quan la UC vulga seleccionar-ho enviarà aquest número a la Unitat de I/O

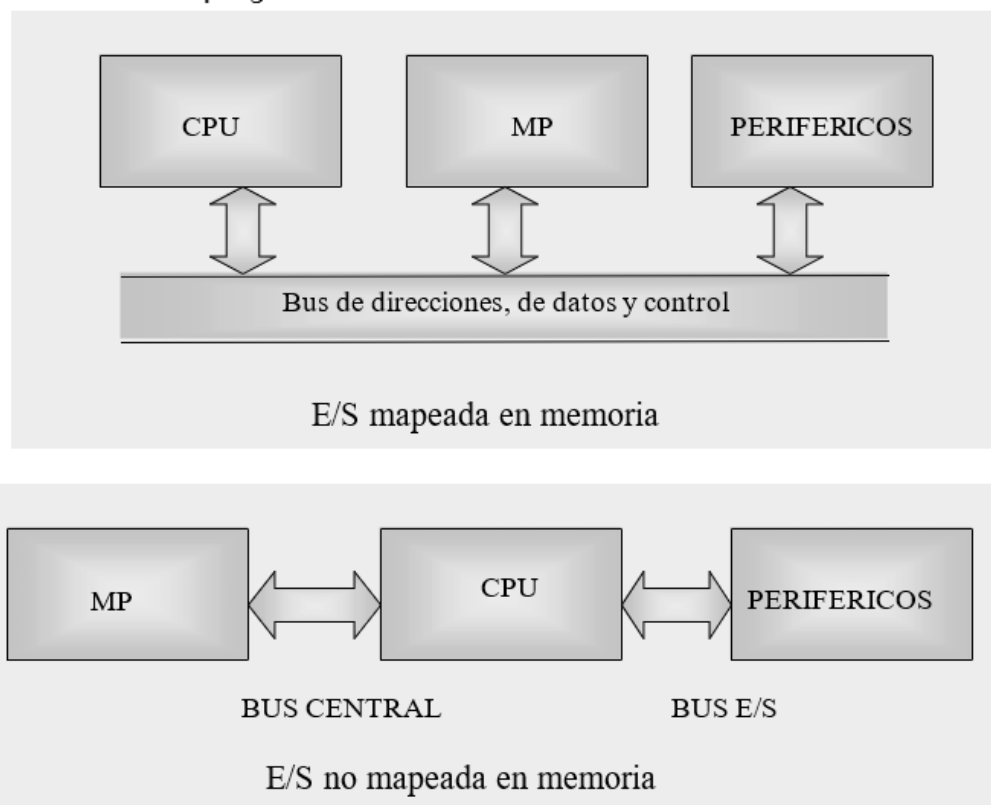
Per a solucionar el problema de la impossibilitat de saber quant temps durara una transferència d'informació amb un perifèric s'han desenvolupat diverses tècniques de comunicació entre la CPU i els perifèrics:

- El més senzill és que la CPU, quan desitge fer una transferència d'informació amb un perifèric, pregunte a la Unitat de I/O si dita perifèrica es troba disponible. Si no l'aquesta, ha de repetir la pregunta una vegada i una altra fins a obtenir una resposta afirmativa, i en aquest cas s'inicia la transferència d'informació.
- Si es desitja obtenir major rendiment de l'ordinador, es pot emprar un altre mètode que es denomina sincronització mitjançant interrupció. La característica d'aquest mètode és que la CPU, en lloc de dedicar-se a preguntar a la Unitat de I/O pel perifèric que desitja utilitzar, el que fa és indicar a la Unitat de I/O que desitja fer una transferència amb el perifèric, i seguidament, si no aquesta el perifèric preparat, comença una altra tasca, oblidant-se momentàniament del perifèric. Quan estiga preparat, la Unitat de I/O indicara a la CPU que pot realitzar-se la transferència; llavors, la CPU interromprà la tasca que estiga realitzant i atindrà el perifèric. D'aquesta manera, la CPU no perd temps esperant al perifèric.

Existeixen diverses maneres d'interconnectar les unitats funcionals de la computadora entre si:

- Una d'elles empra un únic conjunt de busos, al qual es connecten totes les unitats. Això obliga al fet que, en un determinat instant, només dues unitats puguin estar fent ús d'aquest; si cal enviar alguna cosa a la tercera unitat, s'haurà d'esperar que quedi lliure. El principal avantatge d'aquest sistema d'interconnexió és el seu baix cost, i el principal desavantatge, la seua major lentitud. A més, si s'empra el mateix bus de direccions per a indicar la d'un perifèric i la d'una posició de memòria, no podrà coincidir el número de cap perifèric amb cap direcció de memòria; si així ho feren, es produiran errors en seleccionar-se alhora el perifèric i la posició de memòria. Per tant, s'anirà amb compte a assignar als perifèrics direccions que no coincidisquen amb les de la MP. En aquests casos es diu que els perifèrics estan mapatges en memòria, ja que la CPU no distingeix entre les transferències a perifèrics i a MP.

Una altra manera d'interconnectar les unitats de la computadora consisteix a emprar conjunts de busos diferents per a la Unitat de I/O i la MP. D'aquesta manera, encara que s'encareix el sistema, es guanya en velocitat i es poden emprar direccions iguals per a nomenar perifèrics i a posicions de memòria. La CPU, si vol seleccionar un perifèric, utilitza el bus de la Unitat de I/O, i si vol seleccionar una posició de memòria, utilitza el bus de memòria. En aquest cas es diu que els perifèrics no estan mapejats en memòria.



## 2.5. BUSOS DE COMUNICACIÓ

Com ja hem indicat anteriorment la CPU, la memòria i els perifèrics han d'estar íntimament connectades; apareix en aquest moment el concepte de bus: el bus és un conjunt de línies (cables elèctrics) que enllacen els diferents components de l'ordinador, per elles es realitza la transferència de dades entre tots els seus elements.

Un bus de 16 bits transfereix simultàniament aqueixa mateixa quantitat de bits entre dos unitats qualssevol. En funció de la naturalesa o tipus d'informació que aquestes línies transmeten ens trobem amb diferents classes:

- De control: formen part d'ell les línies que seleccionen des d'on i cap a on va dirigida la informació, també les que marquen la seqüència dels passos a seguir per a aquesta transferència. Transmeten informació (senyals) de control, procedent de la unitat de control i el destí del qual pot ser la memòria, per a arrancar cicles de L o E, o qualsevol element d'una altra unitat funcional (perifèrics).
- De dades: per ell, de manera bidireccional, flueixen les dades i instruccions entre les diferents parts de l'ordinador. En un BUS només pot haver-hi una transferència en un instant. Des del BUS es poden carregar simultàniament diversos registres, però només un registre pot bolcar informació al BUS en un instant. La grandària del BUS de dades determina l'ample de paraula
- D'adreces: com vam veure, la memòria està dividida en xicotetes unitats d'emmagatzematge que contenen les instruccions del programa i les dades. El bus de direccions consta d'un conjunt de línies que permet seleccionar de quina posició de la memòria es vol llegir el seu contingut. També

adreça els ports de I/O. La grandària del BUS de direccions determina l'espai d'adreçament de memòria.

### 3 SISTEMES LÒGICS

En general es pot considerar al programari com la part lògica que dota a l'equip físic de capacitat per a realitzar qualsevol tipus de treballs, és a dir, com un conjunt d'idees i processos, desenvolupats per persones, els quals marquen les directrius de treball d'un ordinador. També ho podem definir com el conjunt d'Aplicacions i Programes que permeten operar amb l'ordinador, així com controlar i coordinar els diferents elements del maquinari.

L'element lògic o programari, per a ser present en un sistema informàtic, ha d'emmagatzemar-se en un suport físic: memòria central, memòries auxiliars.

Existeixen diferents tipus de programari:

- Programari bàsic. És el conjunt de programes que l'equip físic necessita per a tindre la capacitat de treballar. Aquest conjunt de programes constitueixen el Sistema Operatiu. Una definició més tècnica ens presenta al sistema operatiu com el suport lògic que controla el funcionament de l'equip físic, ocultant els detalls del maquinari i fent senzill l'ús de l'ordinador a l'usuari.
- Programari d'aplicació. Està format pel conjunt de programes que han sigut dissenyats perquè l'ordinador faci un treball, tant genèric (paquets de programari) com a específic (gestió de nòmines,...) i les dades. Alguns exemples de programari d'aplicació són:
  - APLICACIONS ESPECÍFIQUES: Són aplicacions desenvolupades a mesura de les necessitats de l'usuari.
  - PAQUETS STANDARD: Es compra el paquet i es paga, i ja podem treballar amb el sense cap problema.
  - Llenguatges de programació (entorns de desenvolupament): programes per a desenvolupar altres programes.

#### 3.1. FORMAT DE LES INSTRUCCIONS

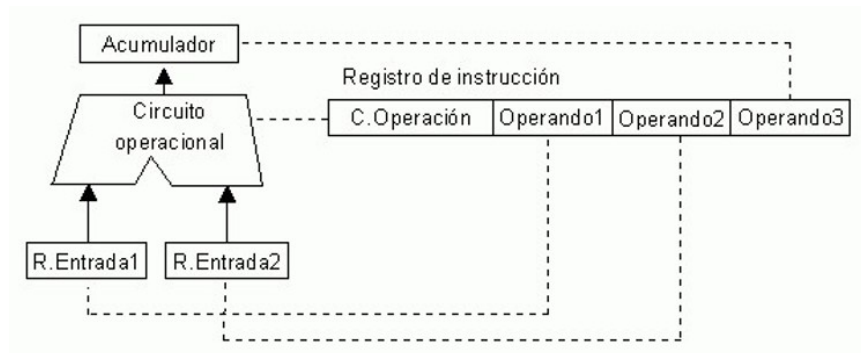
Les instruccions que és capaç de realitzar la UCP es denominen instruccions màquina. El llenguatge que s'utilitza per a la seua codificació és el llenguatge màquina i, d'acord amb la seua funció, es classifiquen com:

- Instruccions de càlcul (aritmètic i lògic) (SUM, RES, MUL, AND, OR, ORX).
- Instruccions de transferència de dades (realitzen el moviment de dades entre la memòria principal i els registres).
- Instruccions de control de flux (Instruccions de salt incondicional o condicional).
- Instruccions d'entrada/eixida (instruccions per a transferir dades des de l'interior de la computadora a un dispositiu exterior o viceversa, així com enviar ordres a aquests dispositius). Etcètera.

Una segona classificació de les instruccions màquina fa referència al seu format i al nombre d'operands que intervenen en elles:

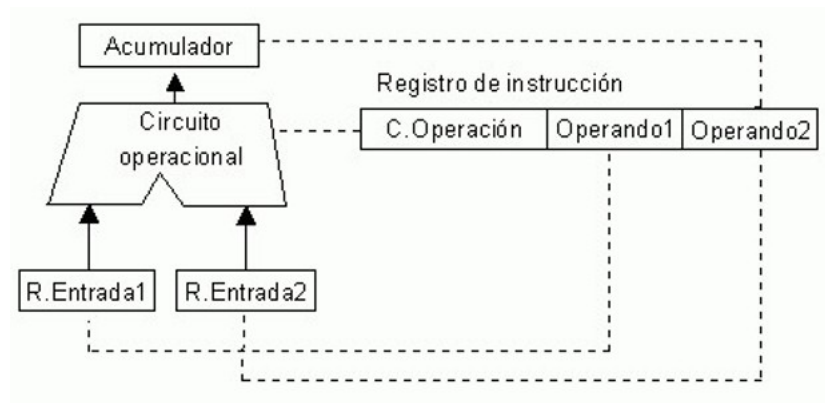
a) Instrucció de tres operands

Està compost per quatre camps. El primer serà el codi de l'operació i a continuació tres operands. Els dos primers són les direccions dels arguments que cal operar i el tercer és la direcció on es depositarà el resultat.



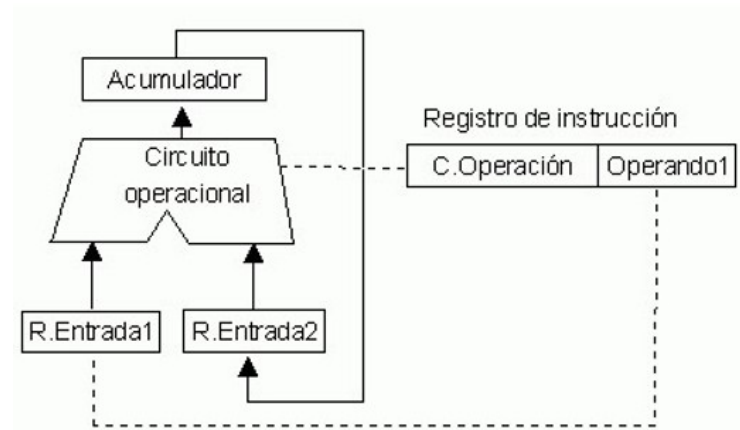
b) Instrucció de dos operands

Conté el codi d'operació i les direccions dels dos operands, dels quals un d'ells actua, a més, com a receptor del resultat de l'operació



c) Instrucció d'un operand

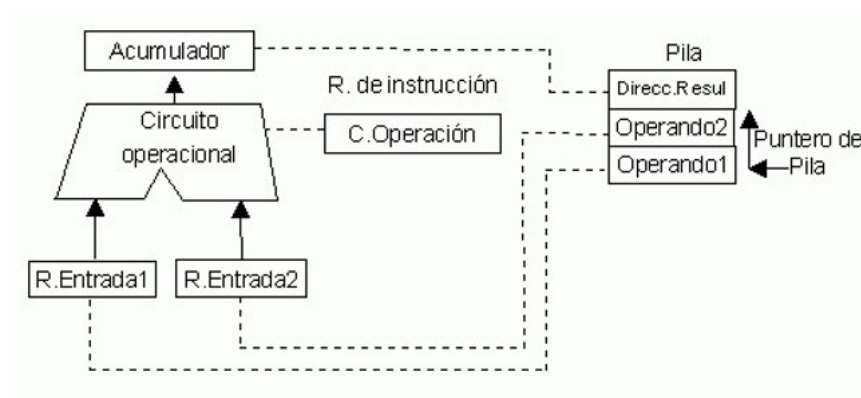
S'utilitza en màquines l'arquitectura de les quals funciona amb filosofia d'acumulador. L'acumulador de la UAL conté prèviament el primer argument de l'operació, el segon és l'indicat en la direcció de la pròpia instrucció, i després de ser operats tots dos pel circuit operacional, el resultat queda depositat de nou en l'acumulador.



#### d) Instrucció sense operands

S'utilitza generalment en computadors l'arquitectura de les quals té filosofia de pila o stack. Una pila està formada per dades emmagatzemades en ordre consecutiu en la memòria, existint un registre especial denominat punter de pila, que ens indica la direcció de l'última dada introduïda en ella. En extraure un element de la pila el punter apuntarà a l'anterior element i anirà disminuint el seu valor fins a arribar al primer element que hàgem introduït.

Aquestes instruccions només tenen el codi de l'operació i si es tracta d'una operació de càlcul extraurà els operands de la pila.



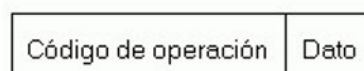
Les computadores, depenent de l'arquitectura que tinguen, poden tindre diversos tipus d'instruccions.

## 3.2. MANERES D'ADREÇAMENT

Hem vist diferents tipus d'instruccions, i en elles sempre hem dit que es troba la direcció de la dada. Ara veurem que diferents maneres existeixen per a adreçar.

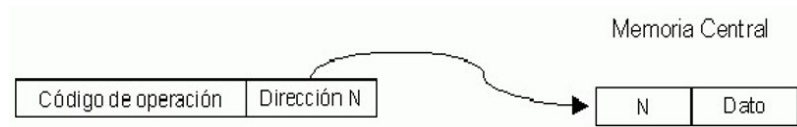
### 3.2.1. ADREÇAMENT IMMEDIAT

En aquesta manera la dada forma part de la instrucció, és a dir, en els operands de la instrucció no es troba la direcció de la dada, sinó la mateixa dada.



### 3.2.2. ADREÇAMENT DIRECTE

La instrucció conté la direcció en la memòria central de la dada que es processarà. Això obliga a haver de fer un accés a memòria per a poder extraure la dada i enviar-lo a la ALU o fins a la unitat designada per la instrucció.

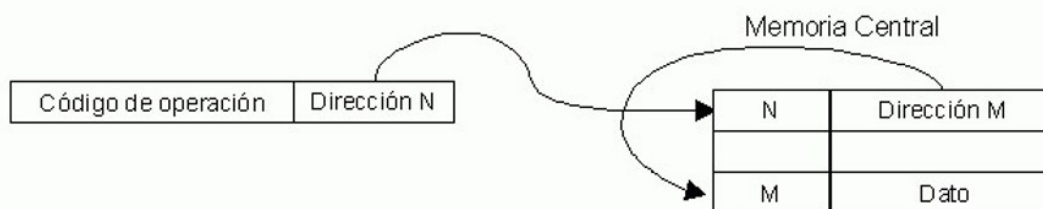


Les característiques principals d'aquest adreçament són les següents:

- No són necessaris càlculs previs per a determinar la direcció final.
- Necessita un accés a memòria (cicle de memòria) per a accedir a la dada, i traslladar-lo fins a la unitat aritmeticològica o fins a la unitat designada per la instrucció.

### 3.2.3. ADREÇAMENT INDIRECTE

Aquest cas és conegut també per indirecció. La direcció continguda en la instrucció no és la de la dada implicada sinó la d'una posició de memòria que conté la direcció d'aqueixa dada. Aquesta posició es denomina direcció intermèdia.



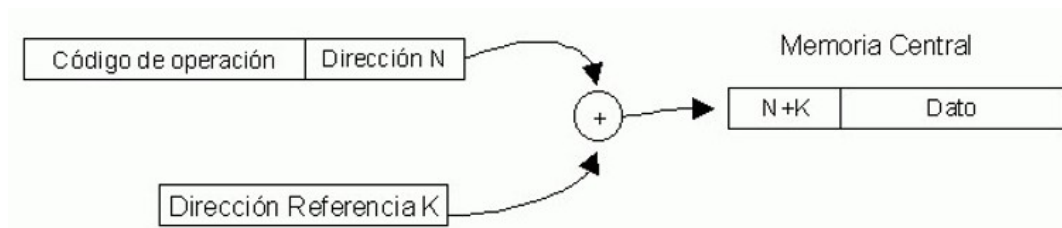
Les característiques principals d'aquest adreçament són les següents:

- No són necessaris càlculs previs per a determinar la direcció final. o Necessita dos cicles de memòria per a accedir a l'operand; durant el primer s'accedeix a la direcció intermèdia i en el segon s'accedeix a l'operand.
- Amb aquest cas s'incrementa notablement la capacitat d'adreçament en poder utilitzar tots els bits de la paraula de memòria.
- La indirecció és un mètode eficaç per a transferir els paràmetres d'un programa principal a subrutines.
- El muntatge d'enllaços en bancs de dades es realitza mitjançant aquest adreçament d'una forma senzilla.
- Aquest adreçament permet amb una sola instrucció, d'un programa, accedir a diferents zones de la memòria.

- Necessàriament, totes les computadores que utilitzen memòria \*paginada necessiten d'un adreçament indirecte

### 3.2.4. ADREÇAMENT RELATIU

En ell, la direcció de memòria on es troba la dada s'aconsegueix sumant la direcció continguda en la pròpia instrucció amb una magnitud fixa continguda en un registre especial. D'aquesta manera es possibilita l'accés a un conjunt de posicions determinades, normalment consecutives, a partir d'una posició considerada com a direcció de referència.



Existeixen més maneres d'adreçament diferents com a resultat de combinar els esmentats anteriorment.

## 3.3. FASES D'UNA INSTRUCCIÓ

Perquè una computadora pugui executar un programa, aquest ha d'estar emmagatzemat en la memòria central. La unitat central de processament prendrà una a una les seues instruccions i anirà fent les tasques corresponents.

Al conjunt d'accions que es duen a terme en la realització d'una instrucció es denomina cicle d'instrucció.

Cada cicle d'instrucció es compon de dues **fases**:

- Fase de cerca o càrrega. En aquesta fase es transfereix la instrucció corresponent des de la memòria central a la unitat de control.
- Fase d'execució. En aquesta fase es realitzen totes les accions que comporta la pròpia instrucció.

### 3.3.1. FASE DE CÀRREGA D'UNA INSTRUCCIÓ.

Suposem que es té un exemple d'instrucció aritmètica de suma amb tres direccions i adreçament directe; és a dir, la instrucció conté el codi d'operació corresponent a la suma, els dos primers operands estan en les direccions de memòria corresponent i el resultat ha de quedar en la direcció indicada pel tercer operand.

SUMAR 033 992 993 (Sumar els continguts de memòria 33 i 992 i emmagatzemar el resultat en la posició 993).

PAS 1.- La UC envia un microorde perquè el contingut del registre de comptador de programa (CP) que conté l'adreça de la següent instrucció (instrucció que correspon processar), siga transferit al registre d'adreça de memòria (RDM). Aquesta informació viatja pel bus de direccions.

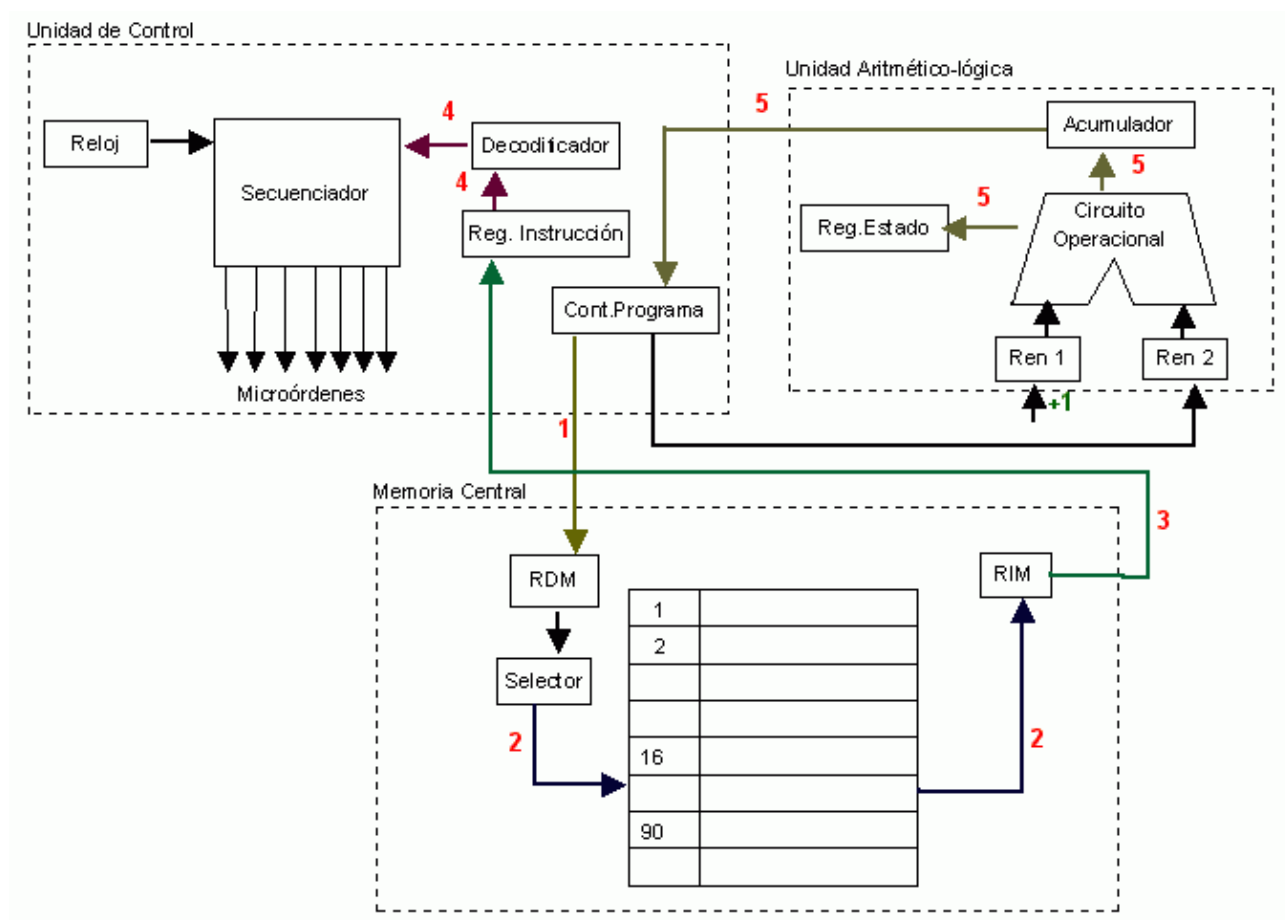
PAS 2.- La posició de memòria que figura en el registre de direcció de memòria (RDM) és utilitzada pel Selector per a transferir el seu contingut (instrucció) al registre d'intercanvi de memòria (RIM).

PAS 3.- Es transfereix la instrucció des del registre d'intercanvi de memòria (RIM) al registre d'instrucció (RI) pel bus de dades.

PAS 4.- A continuació el descodificador procedeix a interpretar la instrucció que acaba d'arribar al registre d'instrucció (RI); en aquest cas SUMAR, quedant disposat per a l'activació del circuit sumador de la UAL i informant el seqüenciador.

PAS 5.- El registre comptador de programa (CP) s'autoincrementa (utilitzant la ALU) amb un valor 1 (o n en el cas que siga aquesta la longitud de la paraula de memòria), de tal forma que quede apuntant a la següent instrucció situada consecutivament en memòria. Si la instrucció en execució és de ruptura de seqüència, el comptador de programa (CP) es carregarà amb la direcció que corresponga.

Quan llig una instrucció de salt a una subrutina, el primer que fa és emmagatzemar l'estat de partida (des de la subrutina que anomena) en el qual es troba, en la zona de stack (pila). Quan trobe la instrucció de retorn llegirà de la zona de stack al fet que punt ha de tornar. Cal dir, que aquesta zona de stack que gestiona pel mètode LIFO (Last In First Out).



### 3.3.2. EXECUCIÓ D'UNA INSTRUCCIÓ.

L'execució es realitza en els següents passos, tenint en compte que si la instrucció no haguera necessitat operands, no s'executarien els passos 1 a 6 ni el 8.

PAS 1.- Es transfereix la direcció del primer operant des del registre d'instrucció (RI) al registre de direcció de memòria (RDM) pel bus de direccions.

PAS 2.- El selector extrau de la memòria aquest dada depositant-lo en el registre d'intercanvi de memòria (RIM).



PAS 3.- S'emporta aquest operand des del registre d'intercanvi de memòria (RIM) al registre d'entrada 1 (REN 1) de la unitat aritmètica. (UAL) pel bus de dades.

PAS 4.- Es transfereix la direcció del segon operand des del registre d'instrucció (RI) al registre de direcció de memòria (RDM) pel bus de direccions.

PAS 5.- El selector extrau de la memòria aquest dada depositant-lo en el registre d'intercanvi de memòria (RIM).

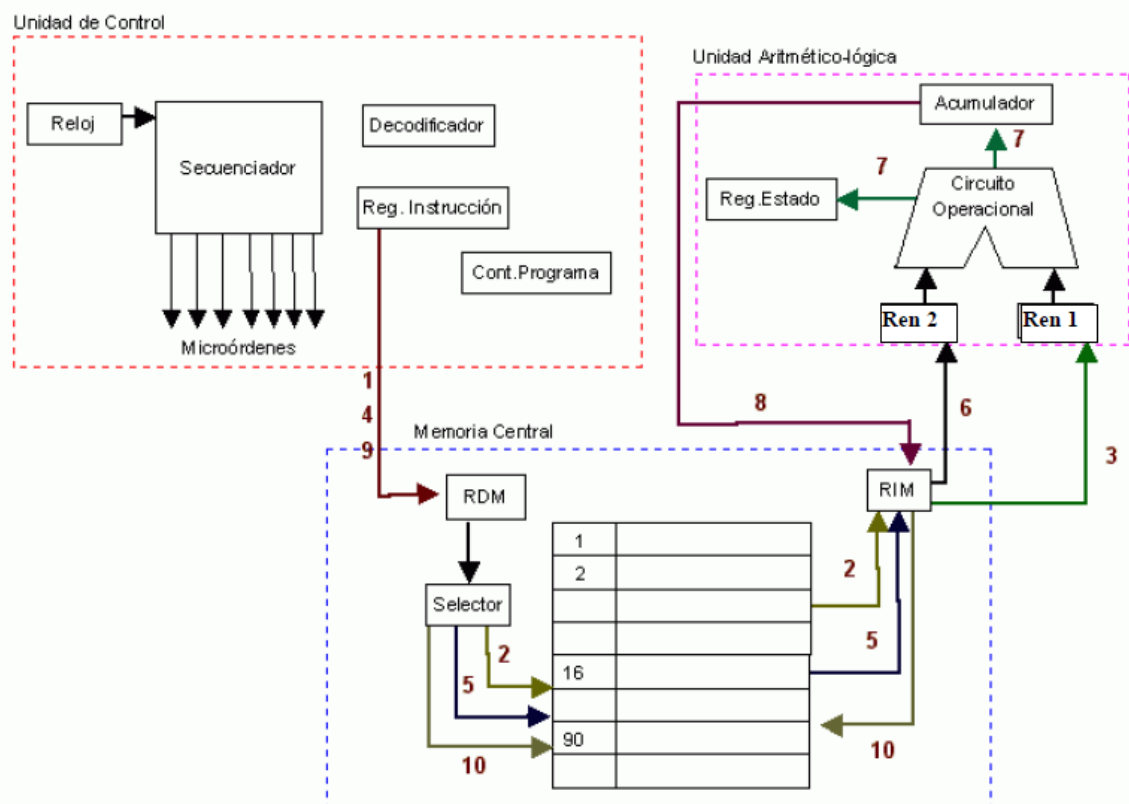
PAS 6.- S'emporta aquest operand des del registre d'intercanvi de memòria (RIM) al registre d'entrada 2 (REN 2) de la unitat aritmètica. (UAL). Per a aquest trasllat s'utilitza el bus de dades.

PAS 7.- El seqüenciador envia un microorde a la unitat aritmètica (UAL) perquè s'execute l'operació de què es tracte. El resultat de l'operació queda emmagatzemat en el registre acumulador (RA).

PAS 8.-Aquest resultat és enviat des del registre acumulador (RA) al registre d'intercanvi de memòria (RIM) pel bus de dades.

PAS 9.- Es transfereix des del registre d'instrucció (RI) al registre de direcció de memòria (RDM) la direcció on ha d'emmagatzemar-se el resultat de la memòria. S'utilitza el bus de direccions per a aquesta comunicació

PAS 10.- Es transfereix el resultat des del registre d'intercanvi de memòria (RIM) a la direcció de memòria indicada en el registre de direcció de memòria (RDM).



Bibliografia: Material específic per als Cicles Formatius d'Informàtica, elaborat per Conveni entre la Conselleria de Cultura, Educació i Esport i la Universitat d'Alacant.