

connecta el micro

5

NOVES TECNOLOGIES

CURS
DE BASIC



connecta el micro

NOVES TECNOLOGIES



FUNDACIÓ CAIXA DE PENSIONS

Edita:
FUNDACIÓ CAIXA DE PENSIONS

President Executiu de la Fundació
Caixa de Pensions:
JOSEP VILARASAU I SALAT

Vice-President Executiu:
RICARD FORNESA I RIBÓ

Director Executiu de la Fundació
Caixa de Pensions:
JOAN JOSEP CUESTA

Cap de Control de l'Obra Social de la Caixa de Pensions:
JOSEP MARIA ARENAS I PASCUAL

Coordinadors de l'edició:
JORDI SALA
ALBERT SÓRIA

Autors:
LLORENÇ HUGUET I ROTGER*, FRANCESC FRANCO I JACOBO, DAMIÀ
CASAS I PESSAFERRER*, MERCÈ GRIERA I FISA*.

Disseny de portada i compaginació:
EQUIP 30/53

Dibuix:
ROGER

Fotografia:
ESPARBÉ-KONIC

Fotocomposició:
CATALANA DE FOTOCOMPOSICION, S.A.

Gestió edició:
MUNDO CIENTÍFICO

Guió de televisió:
JAUME AGUSTÍ I CULELL

Primera edició: juliol 1985
Segona edició: octubre 1985

© LLORENÇ HUGUET I ROTGER, FRANCESC FRANCO I JACOBO, DAMIÀ CASAS I
PESSAFERRER, MERCÈ GRIERA I FISA. 1985.

Tots els drets d'aquesta edició:
FUNDACIÓ CAIXA DE PENSIONS, Via Laietana, 56, 08003 Barcelona

Impressió:
TONSA, Herrera-Alza. Donostia

ISBN: 84-505-1479-7
Dipòsit legal: S.S. 559-85

L'ordinador i

les noves

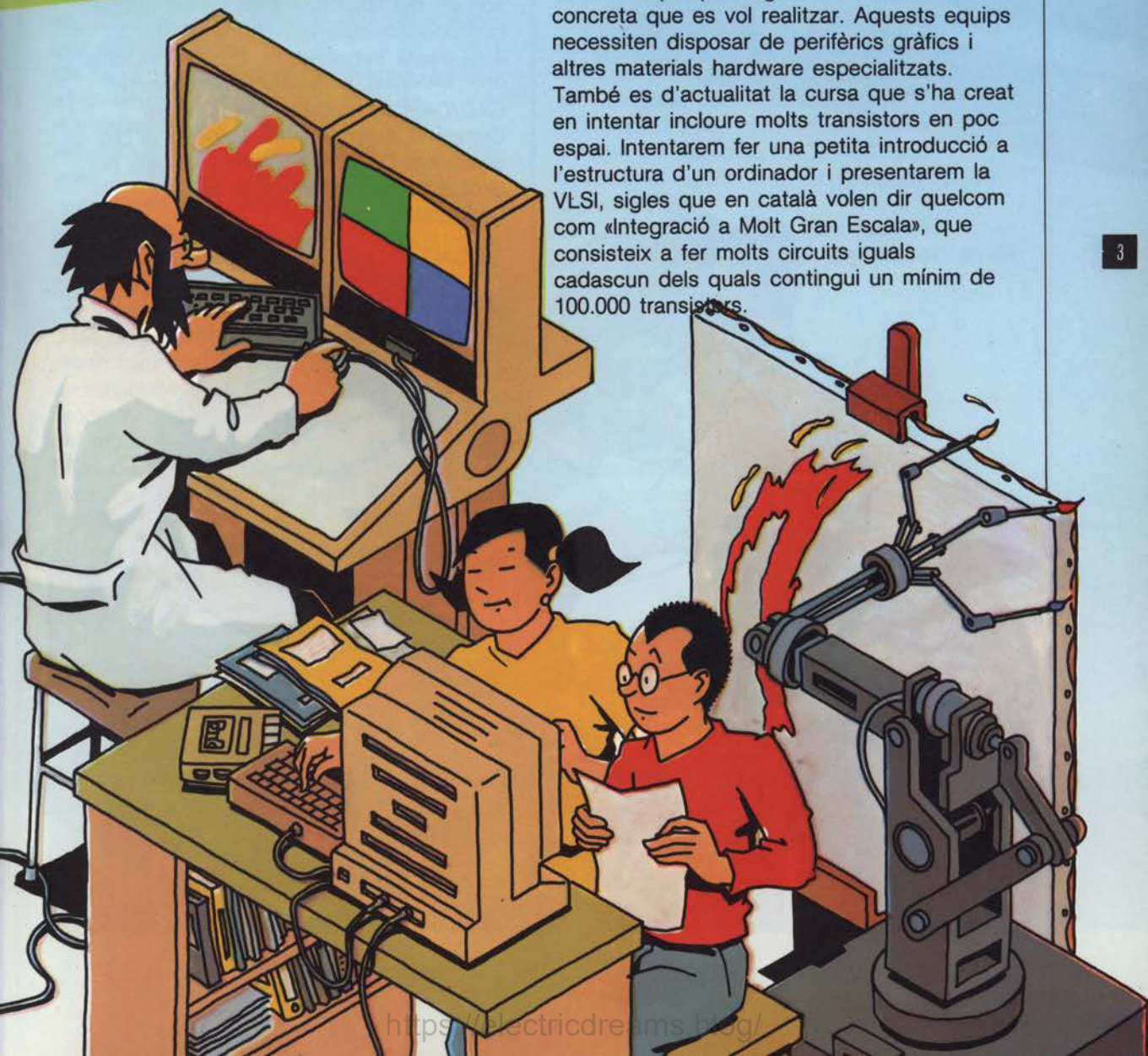
tecnologies

informàtiques

En aquest llibre, a més a més d'oferir-vos un recull del BASIC i un programa en el que tindreu ocasió d'aplicar-les gairebé totes, volem parlar-vos de les innovacions informàtiques que en el nostre país són ja una realitat.

A nivell social, un dels temes dels quals se sent parlar més és el CAD/CAM, que consisteix en una col·lecció de programes capaços d'ajudar al disseny de peces, maquinària i qualsevol tipus de plànols creant-ne a més els models i donant ordres als robots perquè duguin a terme la feina concreta que es vol realitzar. Aquests equips necessiten disposar de perifèrics gràfics i altres materials hardware especialitzats.

També es d'actualitat la cursa que s'ha creat en intentar incloure molts transistors en poc espai. Intentarem fer una petita introducció a l'estructura d'un ordinador i presentarem la VLSI, sigles que en català volen dir quelcom com «Integració a Molt Gran Escala», que consisteix a fer molts circuits iguals cadascun dels quals contingui un mínim de 100.000 transistors.



Disseny

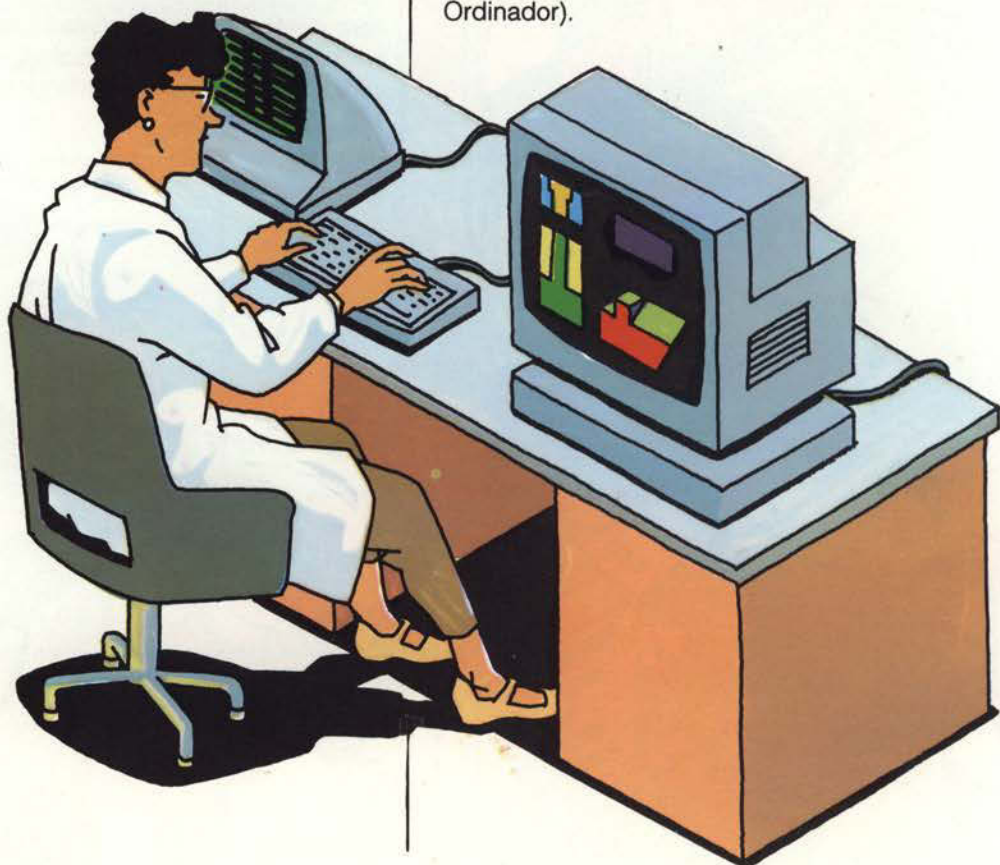
i fabricació

assistits per

l'ordinador

Els sistemes de CAD/CAM

La tecnologia industrial ha trobat en l'ordinador un element molt important per a la concepció, disseny i fabricació de productes. Avui és gairebé impossible pensar en grans plantes de producció on l'ordinador no sigui present. Si en un principi l'ordinador controlava algunes de les màquines d'una cadena de muntatge; ara podríem dir que ha pres la iniciativa de dissenyar els productes que ell mateix pot fer i fabricar. Tot això, evidentment, conduït per la mà d'experts que són els qui faran possibles certes tasques que podríem qualificar d'intel·ligents. Avui s'està treballant, en el món industrial, en sistemes informàtics denominats CAD/CAM (Computer Aided Design/ Computer Aided Manufacturing) o el que és equivalent DAO/FAO (Disseny Assistit per Ordinador/Fabricació Assistida per Ordinador).



Un sistema CAD/CAM reposa sobre un paquet de programes (software) i més o menys, en les prestacions específiques que poden realitzar un conjunt de perifèrics al voltant d'un ordinador estàndard (hardware). Es pot pensar que el CAD/CAM és un sistema informàtic a cavall de la robòtica, l'enginyeria i la intel·ligència artificial. Parlem primer del CAD. Un software especialitzat en CAD és un paquet de programes interactiu (home-ordinador) que ajuda al disseny d'un projecte d'enginyeria mitjançant unes funcions (subrutines) específiques que faciliten el seu dibuix sobre una pantalla gràfica.

Un cop fet el disseny del prototipus desitjat s'han de poder realitzar una colla de manipulacions sobre el dibuix, com ara: reduccions a escala, ampliacions d'una certa zona, ..., fins i tot gravar les comandes necessàries per al control automàtic de la seva producció.

La CAM és la continuació lògica del CAD quan es disposa d'una instal·lació de producció automatitzada mitjançant màquines dites de **control numèric** (màquines que reben les instruccions de control a través de codificacions digitals, normalment programables automàticament) que poden ser robots.

Si es disposa del procés CAD/CAM, aleshores el prototip concebut des del CAD pot ésser directament fabricat mitjançant la CAM quan es disposa d'eines modulars i programables, és a dir, de màquines

capaces de fabricar diferents peces que obeeixen unes comandes, que generalment es reduiran a una seqüència d'ordres del tipus: avançar 2 mil·límetres i posar una soldadura, girar 35 graus i avançar 4 centímetres, aleshores fer un forat de 6 mil·límetres de diàmetre, etc.

El sistema CAD/CAM tingueren el seu començament els anys seixanta i a final d'aquella dècada ja eren utilitzats en les indústries electròniques de fabricació de circuits impresos. Posteriorment, a mitjan anys setanta, quan arreu del món es plantejà el problema de la producció industrial, els Estats Units començaren a fomentar la recerca en el que s'ha denominat les **altes tecnologies**, com a solució al problema esmentat. Es així com a final dels anys setanta els sistemes CAD/CAM foren utilitzats en aplicacions mecàniques, civils i d'arquitectura.

En l'actualitat s'utilitzen tècniques de CAD/CAM en el disseny i la producció en els sectors:

- tèxtil
- de sabates
- mecànic
- de mobles
- electrònic
- metal·lúrgic
- de xarxes de distribució de fluids
- de prospeccions petrolíferes
- d'anàlisi d'estructures
- de programes de robots
- ...

Disseny assistit per ordinador (CAD)



Fabricació assistida per ordinador (CAM)





Estació de treball: Pantalla gràfica i taula digitalitzadora

i comencen a penetrar com a ajut en el disseny de plànols d'arquitectura, en cartografia, en impremtes i gabinets de publicistes. És el que s'anomena una tecnologia punta.

Configuració bàsica d'un sistema CAD/CAM

Els sistemes de processament d'informació interactius, com és el CAD/CAM, estan dissenyats bàsicament per poder treballar en forma de diàleg entre home i ordinador. L'usuari fa saber les seves intencions a l'ordinador mitjançant unes comandes i aquest, un cop processada la informació, dona resposta a l'usuari. El fruit d'aquests intercanvis d'informacions, en molts casos visuals, serà el disseny del prototip que l'home ha concebut.

Els sistemes CAD/CAM són compostos per tres unitats funcionals:

- estacions de treball
- estacions de producció
- processador gràfic

l'estació de treball: és el lloc on l'enginyer pot «conversar» amb l'ordinador, entrant i traient models, dibuixant informacions,



manipulant traçats geomètrics, demanant pels càlculs efectuats, etc... De fet, la majoria dels sistemes permeten que totes les funcions del CAD/CAM puguin fer-se des de l'estació de treball, que es converteix així en el punt important de la interacció home-ordinador.

En general, les estacions de treball són compostes de dispositius d'entrada i sortida (teclats, taules digitalitzadores, funcions de menús mitjançant cursors, pantalles d'alta resolució,...) així com, a vegades, de processadors locals gràfics i de propòsit general.

l'estació de producció: és la que genera les sortides del sistema CAD/CAM tals com traços, informacions de models geomètrics, cintes de control numèric, llistes de materials, fitxers de dades digitals, etc, utilitza plotters, impressores, disquets, cintes... Cada tipus d'estació de producció produeix una classe específica de sortida.

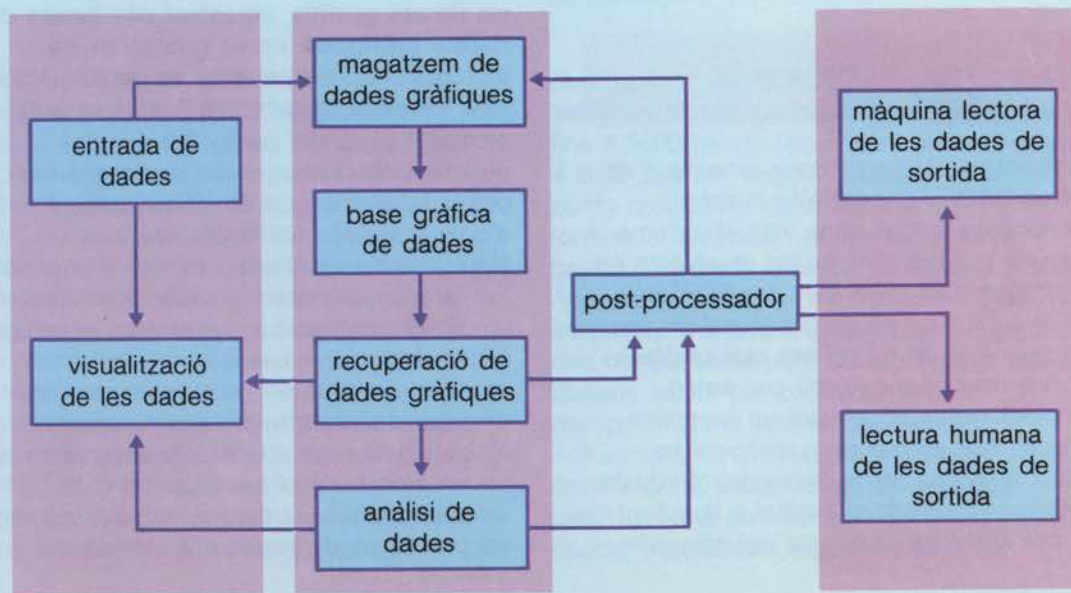
el processador gràfic: és la part més important, des del punt de vista informàtic, d'un sistema CAD/CAM i és constituït per l'ordinador principal, la memòria central, l'emmagatzematge en línia (disc), l'emmagatzematge fora de línia (cintes magnètiques) i el corresponent software. Els sistemes CAD/CAM estan configurats de manera que les estacions de treball van



connectades bidireccionalment al processador gràfic per transmetre les transaccions d'entrada i sortida entre ambdós, mentre que les estacions de producció només són connectades de manera unidireccional, ja que són dispositius només de sortida. És a dir, el processador és el que transfereix informació a les estacions de producció, però no en rep d'aquestes.

Quan es disposa de màquines de control numèric, entre el processador gràfic i l'estació de producció es col·loca el que se'n diu **post-processador**, el qual s'encarrega d'interpretar les sortides de control numèric per fer-les compatibles amb cada màquina particular. Podríem dir que el post-processador personalitza amb la màquina que ha d'efectuar el treball corresponent al disseny.

organització funcional d'un CAD/CAM



estació de treball

processador gràfic

estació de producció



Dispositius de sortida d'un sistema CAD/CAM



Perifèrics d'un sistema CAD

Software d'un CAD/CAM

El software és la part menys tangible d'un sistema DAO/FAO però n'és la més important: és la que dóna vida al sistema. A causa de la seva complexitat es realitza en llenguatges d'alt nivell com ara FORTRAN, PASCAL, PL/1... i és desenvolupat en una gran quantitat de subrutines que realitzen funcions específiques però que estan interrelacionades mitjançant un programa principal. Així, el programa principal es redueix a reconèixer la comanda introduïda per l'usuari i a cedir el control a la subrutina, que pot entendre i executar aquella ordre.

Podríem dir que el programa principal és un menú. Aquesta manera d'organitzar el software, a part d'ésser una bona forma de realitzar programes de grans dimensions, dóna molt joc a l'hora de voler introduir noves subrutines que facin altres funcions. Cadascuna d'aquestes subrutines realitza una funció predeterminada, per exemple traçar una línia o un arc entre dos punts, fer un ombrejat d'una àrea concreta, dibuixar un polígon que passi per un cert nombre de punts, rotar els eixos de coordenades, poder veure la figura dibuixada des d'una altra perspectiva, fer un «zoom», deduir la planta, alçat i perfil d'una figura...

En aquest tipus de software són fonamentals els **fitxers gràfics**, altrament dits **bases de dades gràfiques**, on es guardaran els dibuixos com una col·lecció de coordenades amb els seus atributs: arc (x,y), línia (y,z), ploma (j), ombrejat (x+3,y-2)... La gran quantitat d'ordres que pot implicar un sol dibuix fa pensar que és molt important d'optimitzar aquests fitxers utilitzant tècniques de compactació de la informació. Per la gran quantitat de dades que comporta un dibuix processat amb aquesta tecnologia, a part de requerir bases de dades jeràrquiques o bé en xarxa, es necessita disposar d'elements que siguin capaços d'accelerar l'accés al disc. Aquests sistemes poden arribar a multiplicar per cinc la velocitat d'accés al disc si es comparen amb els processos de dades alfanumèriques.

ARCHITECTURAL PLANS

The menu is divided into several sections:

- DOORS:** Includes options like DELETE DOOR, MOVE DOOR, and various door types.
- WINDOWS:** Includes DELETE WINDOW, MOVE WINDOW, and window types.
- STRUCTURAL GRIDS:** Includes AUTO FULL MATRIX, REPLACE COLUMN, PLACE COLUMN, COPY GRID, and SINGLE TAG.
- TOILETS:** Includes HANDICAP RAILS and PARTITION DOORS.
- SINKS:** Includes options for sink placement.
- NOTES:** Includes TEXT, TEXT TEXT, and SCALE.
- SCHEDULES:** Includes EDIT and ATTACH.
- MISC.:** Includes PATTERNS, SCALE, ANGLE, and DIMENSIONS.
- FILES:** Includes OPEN, SAVE, PRINT, and other file operations.
- SCREENS:** Includes L, R, BOTH, VIEW, STOP DRAWING, VIEW ON, VIEW OFF, ZOOM IN, ZOOM OUT, FIT, AREA, CENTER.
- ANGLE INPUT:** A circular diagram for selecting angles.
- DIMENSION INPUT:** A grid for entering dimension values.
- GRID:** A grid for setting grid dimensions.
- USER INPUT:** Includes AX, AY, AREA, DEFINE, ELEMENT.
- ELEMENT STYLES:** A table for changing weight, code, and color.
- GRAPHIC ELEMENT PLACEMENT:** A table for placing various shapes like LINE, ARC, ELLIPSE, CIRCLE, etc.
- GRAPHIC ELEMENT MANIPULATIONS:** Includes DELETE, MOVE, COPY, MODIFY, SCALE, ROTATE, PARTIAL DELETE, and COPY PARALLEL.
- DIGITIZER:** Includes SET UP and PLOT.
- FENCE:** Includes MOVE, COPY, SCALE, ROTATE, DELETE, LOCKS, and UNLOCK.

Menú pel disseny automatitzat de plànols d'arquitectura amb un sistema CAD

També és important, a l'hora de veure les possibilitats del CAD/CAM, ressenyar la tendència actual a les xarxes de teleprocés. Els punts d'aquesta xarxa poden ésser l'ordinador que porta tota la gestió del CAD/CAM o qualsevol perifèric. Podríem pensar en una fàbrica que disposa del processador central i alguns perifèrics a la seu principal i en cada sucursal té terminals gràfics amb certa intel·ligència. La connexió per teleprocés estalvia repetir informacions, car es disposarà d'una sola base de dades gràfica, i es poden compartir recursos. No oblidem, però, que una xarxa de teleprocés està comunicada per fil telefònic, tant si utilitzen les línies de lloguer punt a

punt com si utilitza la RAC. Ambdós sistemes només permeten una transmissió fins a 9600 bauds (equivalent en molt casos a 9600 bits per segon) utilitzant entre dos punts qualsevols de la xarxa (ordinador-ordinador, ordinador-terminal gràfic,...) modems clàssics com ara el RS232. Aquesta rapidesa de modulació és clarament insuficient per poder treballar en temps real des de qualsevol punt de la xarxa, ja que el disseny assistit per ordinador implica manipular grans volums de dades. Actualment, la C.T.N.E. té, a títol experimental, línies de transmissió que permeten una comunicació de fins a 20.000 bauds, però el seu lloguer és molt car.

Digitalitzador



Scanner



Estació de treball



Dispositius d'entrada d'un sistema CAD/CAM

Hardware d'un CAD/CAM

El hardware d'un sistema CAD/CAM consisteix en un ordinador central: un VAX, o un IBM, o un HP, o un Cyber, o qualsevol altre ordinador mitjà o gran, que generalment treballen amb paraules de 32 bits en lloc de vuit com és el cas dels microordinadors que fins ara utilitzem, i un conjunt de perifèrics de propòsit especial per al disseny i tractament d'informacions gràfiques.

Pel que fa referència al hardware d'entrada d'informació, podem destacar els terminals de propòsit general, com el teclat d'un ordinador qualsevol, i els digitalitzadors. El primer ens servirà per donar entrada als valors alfanumèrics de la informació a processar mentre que usarem el segon per actuar «virtualment» sobre la pantalla, entrant aquelles ordres gràfiques que calguin, com ara traçat de corbes o ombrejats. Al mateix temps els valors numèrics d'aquestes accions quedaran guardats en memòria.

La utilització del digitalitzador es fa, generalment, amb una taula de menús i amb l'ajut d'un cursor òptic. Aquesta taula de menús porta tots els elements gràfics necessaris com ara: definir el sistema de coordenades, unir dos punts amb un arc, traçar una recta que passi per un cert punt i amb un cert angle, esborrar qualsevol informació entrada equivocadament ... Totes aquestes accions que nosaltres podem entrar de manera directa són interpretades pel

programa principal, que passa el control a la subrutina corresponent. Normalment aquestes subrutines han estat programades utilitzant propietats geomètriques i funcions especials de la trigonometria. És en els dispositius de sortida on hi ha la gamma d'elements més sofisticats. Els més importants són les pantalles gràfiques i els plotters. Des dels anys cinquanta que aparegué el primer ordinador capaç de controlar un tub de raigs catòdics s'ha anat progressant en aquest camp molt ràpidament i avui podem trobar pantalles gràfiques de diferents tipus: d'emmagatzematge, de refresc, de raster... que utilitzen el fòsfor com a element visualitzador un cop atacat per un raig d'electrons. Actualment ja s'està treballant amb les pantalles de «cristall líquid». Si avui els monitors gràfics dels microordinadors personals poden assolir nivells de resolució de 250.000 punts i vuit colors, les pantalles d'alta resolució utilitzades en sistemes CAD/CAM poden arribar a tractar fins a tretze milions de punts i més de quatre mil colors.

Una **pantalla gràfica** és igual que una pantalla de les anomenades alfanumèriques (les de televisió o les dels micros), però compta amb tres característiques fonamentals:

- els punts queden perfectament definits, no hi ha difuminació al seu entorn. Això depèn de l'amplada d'ona del raig d'electrons i de la qualitat del fòsfor.

-nivell de resolució molt elevat, unes 4.000 línies.

-tenen capacitat de memòria.

Quant a la velocitat de comunicació entre ordinador i pantalla gràfica, direm que aquesta pot ésser fins a cinc vegades superior que una comunicació ordinador-pantalla alfanumèrica.

Pel que fa referència als plotters, hem de dir que també són indispensables en un sistema de CAD/CAM ja que són els que permetran de treure sobre paper el resultat del disseny concebut per l'enginyer i realitzat per l'ordinador. Aquests es divideixen en dos grans grups; els plotters de plomes i els electrostàtics.

Els plotters de plomes són capaços de realitzar un dibuix mitjançant el desplaçament d'una ploma, retolador o qualsevol eina de dibuix, sobre un suport de paper. En aquest cas el dibuix és realitzat traçant una successió de rectes de llargada variable seguint uns eixos de coordenades. És a dir, hi ha quatre direccions fonamentals; anar a dreta, a esquerra, a dalt i a baix, la qual cosa normalment s'expressa per eix X, (+X,-X) i eix Y (+Y,-Y). L'eina de dibuix utilitzada podrà prendre només dues posicions: la de contacte amb el paper i la de no contacte. La combinació de les successions de rectes en la direcció dels eixos de coordenades i el contacte o no de l'eina amb el paper, ens permetrà de realitzar



qualsevol dibuix. La perfecció del resultat dependrà del nivell de resolució de l'eina utilitzada i del desplaçament elemental en la direcció dels eixos.

En un plotter de plomes es té en compte la velocitat de dibuix, mesurada en centímetres per segon, el nombre de plomes que pot utilitzar simultàniament i el format del paper que pot utilitzar. En el cas dels sistemes CAD/CAM s'utilitzen plotters de 8 plomes i amb una velocitat de dibuix que oscil·la entre els 60 i 90 centímetres per minut sobre un paper DIN A0.

Quant als plotters electrostàtics, tenen un funcionament molt diferent que els de plomes. En aquest cas no hi ha pas desplaçament de l'eina de dibuix, sinó que s'utilitza una línia d'agulles situada al llarg del paper, que és el que es mou i només en una direcció. El dibuix es va realitzant línia a línia gràcies a les agulles, que funcionen com si fossin electrodes creant càrregues electrostàtiques sobre cada punt del paper. En ésser tractat aquest amb una tinta especial, només queden impregnats aquells punts que han rebut la càrrega electrostàtica. La resolució d'un plotter electrostàtic depèn del nombre d'agulles que té per centímetre (o bé per polzada). Les característiques a tenir en compte en un plotter electrostàtic són el nombre de línies per minut que és capaç de dibuixar, la resolució, que es mesura pel nombre de punts per mil·límetre, i les mides del paper que pot utilitzar.

Un tipus de pantalla gràfica





Una aplicació de CAD en arquitectura

Quan hom pensa en disseny, possiblement li ve al cap un estudi d'arquitectura tot farcit de grans taules de dibuix on els arquitectes i els delineants intenten donar forma a una idea. El seu treball consisteix a fer els plànols per a la construcció d'un edifici, tenint en compte els factors econòmics, socials i de gust del client.

Nosaltres hem tingut la possibilitat de conèixer el gabinet d'arquitectura de la casa INTERGRAPH (Interactive Graphics). Disposa d'un sistema CAD especialitzat en arquitectura, que presentem breument.

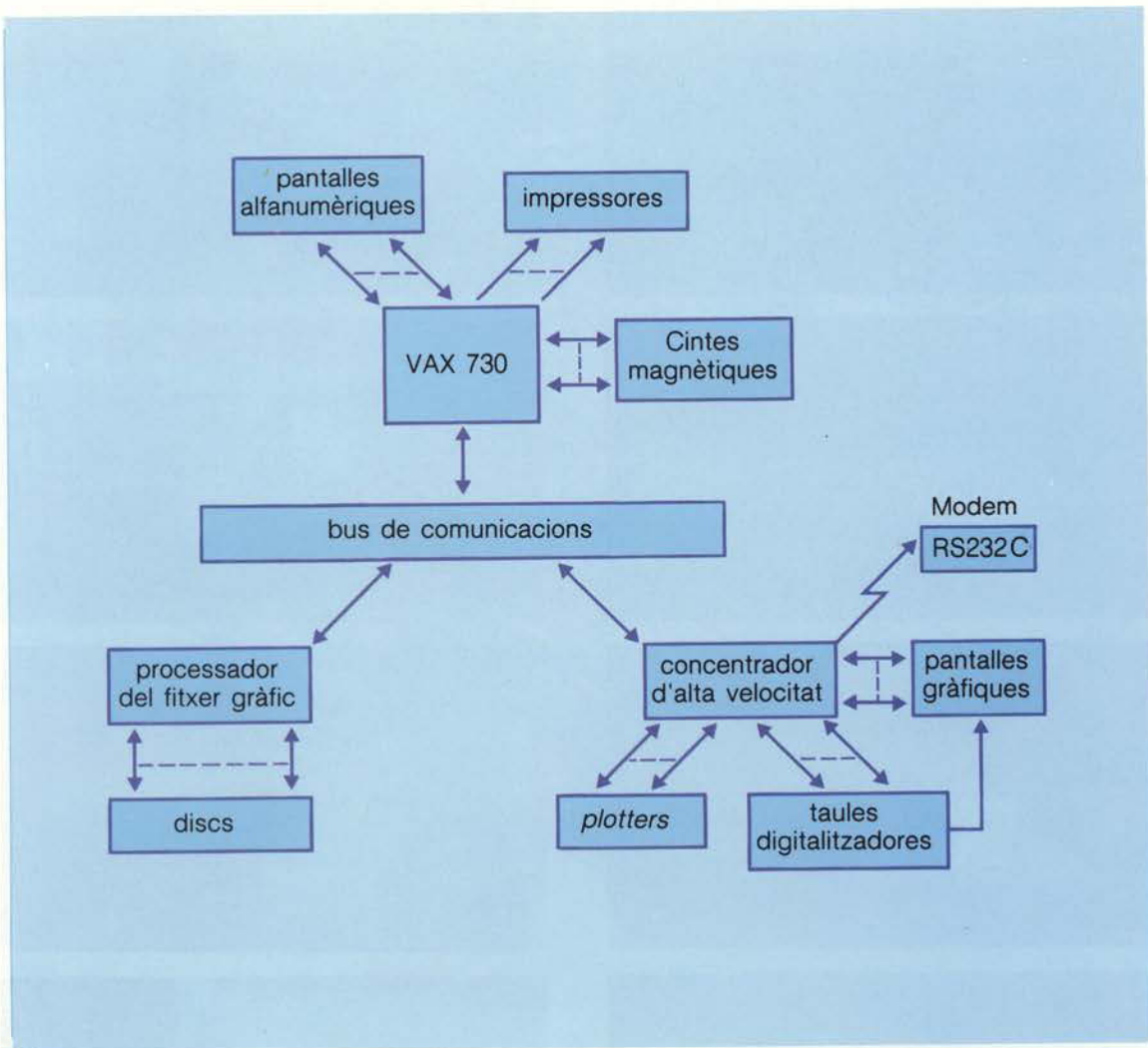
Com tota opció CAD, aquest és un sistema que permet una total interacció gràfica entre l'arquitecte i l'ordinador mitjançant les tauletes digitalitzadores i la utilització de menús gràfics per a cada operació:



Aspectes d'un disseny arquitectònic

planificació dels espais, modelatges d'interior i d'exterior, càlculs d'àrea, distribució de les xarxes de serveis...

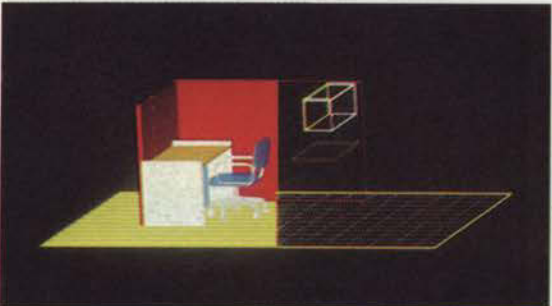
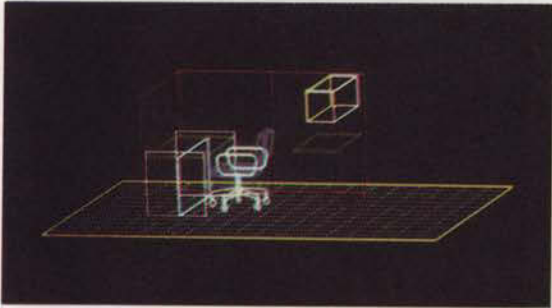
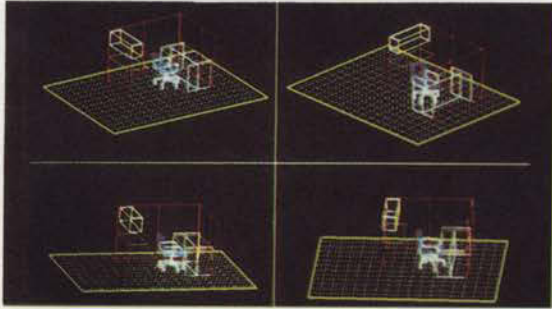
Un estudi d'arquitecte basat en un sistema CAD podria tenir la següent configuració:



L'arquitecte, assegut davant la pantalla gràfica, pot entrar el seu disseny a través d'un cursor que es mou sobre una taula digitalitzadora marcant cadascun dels elements bàsics que vol utilitzar, segons un menú especial, i decidint a quin lloc de l'espai el vol col·locar. Primer entraria aquells elements definidors del contorn i podria visualitzar diferents perspectives del seu disseny. En segon lloc pot donar color i ombrejar com ell vulgui les diferents zones dibuixades. Un cop aconseguit el primer mòdul pot anar ampliant el seu disseny

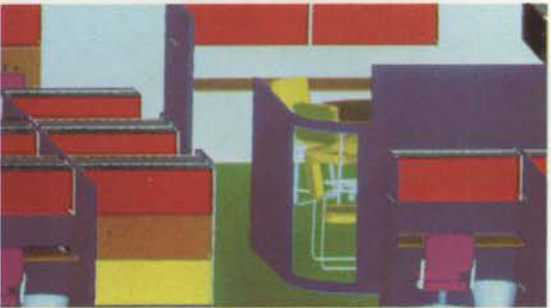
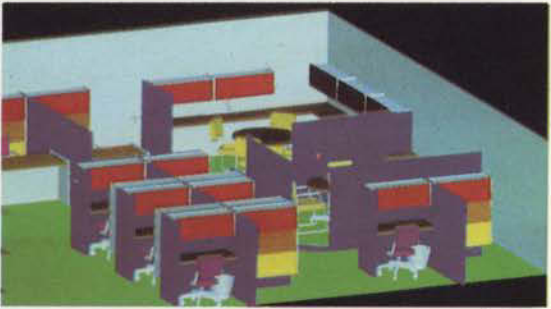
inicial fent més mòduls que, un cop acabats, podrà veure en forma conjunta.

Si n'està satisfet pot emmagatzemar aquest dibuix en el seu fitxer o base de dades gràfica. Quan arribi el client li podrà ensenyar el disseny global i podrà realitzar «zooms», perspectives, canvis de color, canvis de components... Si cal alguna modificació, aquesta serà feta en presència del mateix client i s'actualitzarà el fitxer gràfic. Un cop donat el disseny per bo ja pot començar a treure'n tots el plànols que li facin falta: planta, alçada i perfil, amb les corresponents mides.



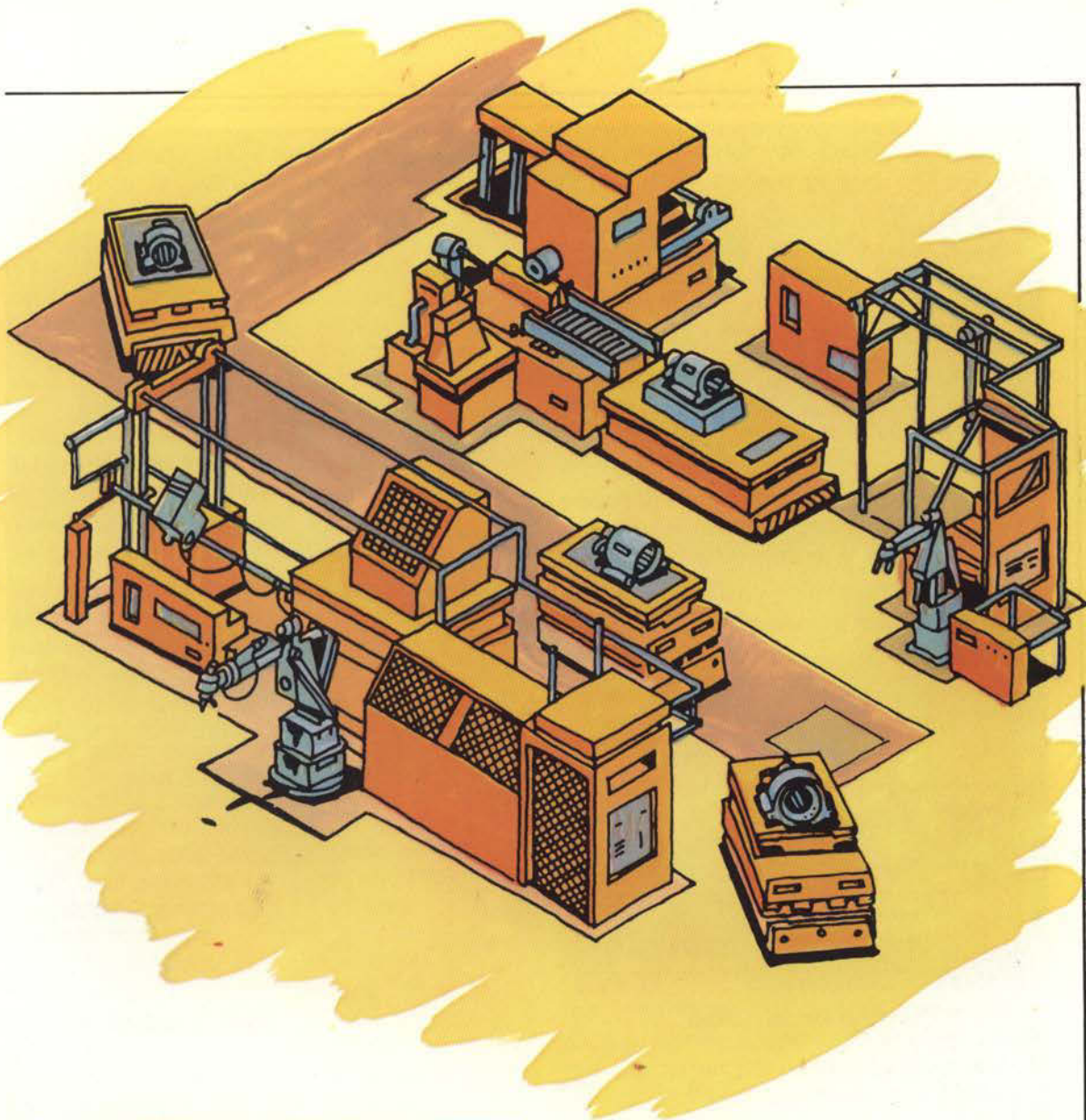
Seqüència d'un disseny en arquitectura

A més, utilitzant tota la informació de què disposa podrà fer el plec de condicions tècniques que ha d'adjuntar per poder dur a terme l'obra, com ara càlcul de les estructures, disposició de les xarxes de



Manipulació del objecte dissenyat (ZOOM)

distribució de llum, gas i aigua. Un cop més cal observar que l'ordinador no ve a substituir la capacitat creativa de l'arquitecte, sinó que ve a ajudar-lo en la part mecànica de disseny del seu projecte.



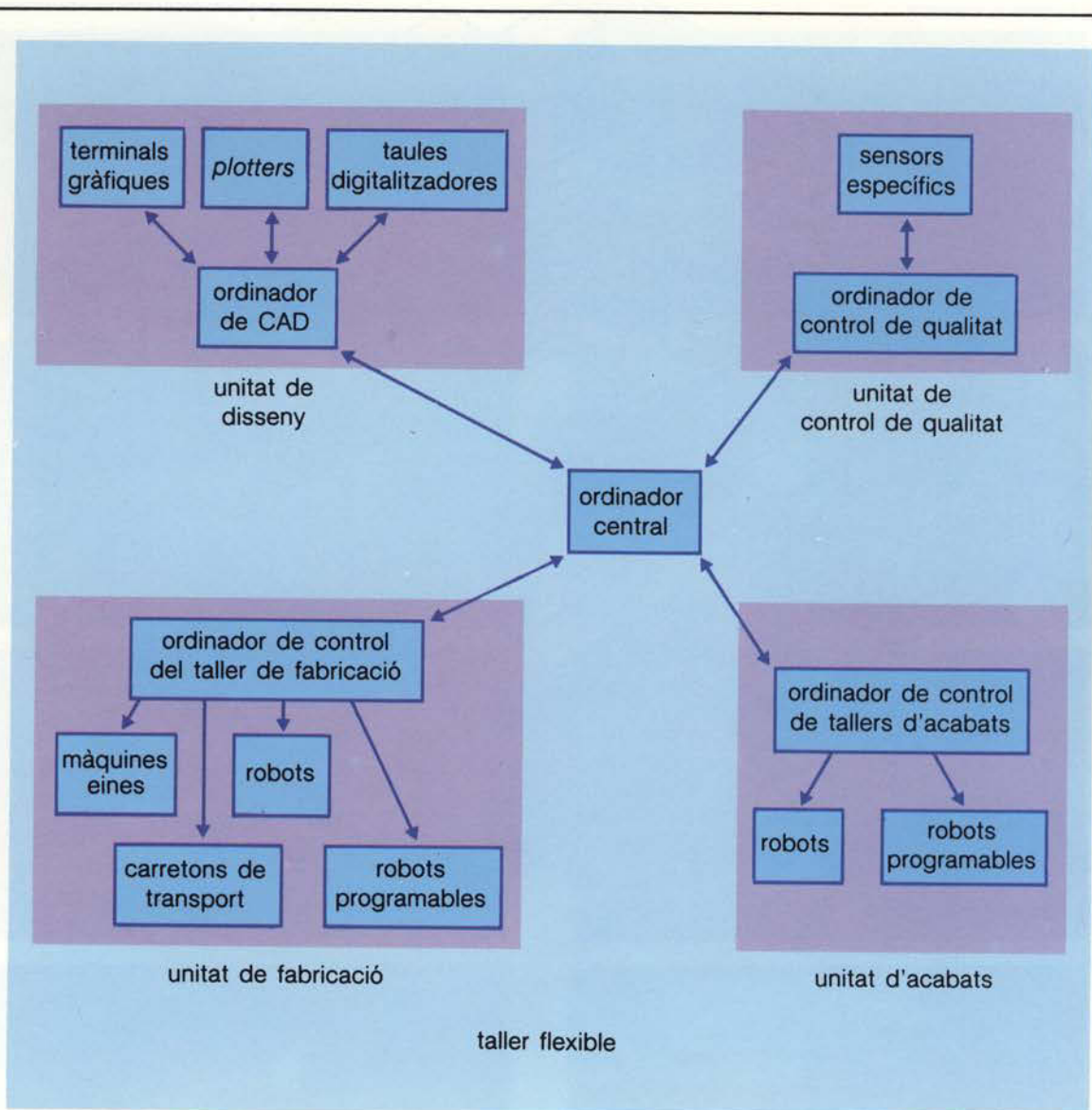
El taller flexible

Un taller informatitzat, és a dir, equipat amb robots, és capaç de fabricar en sèrie un gran nombre de peces, però totes iguals, amb les mateixes característiques. Canviar de model suposa una feina de readaptació molt gran i a vegades impossible.

Els **tallers flexibles** presenten l'oportunitat de dissenyar peces noves (prototips) que ni els qui les han inventades coneixen. Quan un

enginyer concep la peça amb l'ajut d'un ordinador equipat amb tecnologia de CAD les característiques d'aquesta passen automàticament al taller, la peça es fabrica i al cap de poc temps l'enginyer en pot tenir unes quantes a les seves mans per dir si li semblen bé o no i què cal fer-ne.

Un taller flexible podria representar-se pel següent diagrama de blocs:



L'enginyer dissenya el seu model de peça usant la tecnologia i programes de CAD que té el seu abast. Les característiques d'aquest nou producte són enviades directament a l'ordinador central. Aquest dona ordre a l'ordinador del taller de la fabricació per tal que se'n faci una de mostra. Aquest ordinador comanda totes les màquines del taller i fa que aquestes construeixin físicament una unitat del que se

li demana. Aquesta, un cop construïda, s'anomena **peça bruta**.

La peça bruta viatja sobre un **carretó** (robot mòbil) cap a la unitat d'acabats, allà es neteja de tota impuresa passant per una sèrie de robots programables que fan feines com ara llimar, rentar, eixugar...

Quan la unitat d'acabats té la feina feta ho comunica a l'ordinador central, que activa la unitat de control. La peça ja neta arriba

sobre un carretó i se la sotmet a diverses proves: pes, grandària, formes, resistència del color... Els resultats d'aquestes són processats per l'ordinador de control i s'envien a l'ordinador central. Aquest ordinador compara aquestes dades amb les dades teòriques que li van arribant de la unitat de disseny. Si la peça no ha quedat prou bé la fa tornar a fabricar els cops que calgui fins que sigui tal com ha d'ésser. Aleshores dona ordre a la unitat de fabricació per tal que en faci al voltant d'una dotzena que, un cop passades la neteja i el control de qualitat, arriben a mans de l'enginyer perquè en doni la seva opinió:

-cal retocar quelcom en el disseny i tornar a refer el procés.

-cal muntar-la en una màquina per veure si dona el rendiment esperat.

-s'ha de fabricar en sèrie.

-serà un prototip per presentar en una fira.

.....

Tot aquest procés no té cap intervenció manual. L'home només la dissenya, la peça «es fa sola». Crear un prototip que en un taller flexible suposa unes tres setmanes significaria quatre mesos en un taller habitual.

En un taller flexible és l'ordinador qui, treballant en temps real, gestiona tota la producció:

- **optimitza el temps de funcionament** de cada màquina. En un taller normal la maquinària treballa un 30 % del temps de funcionament del taller mentre que en un de flexible treballa el 80 % del temps.

- **aprovisiona** els robots dels instruments (claus algheses, tornavisos, alicates...) i matèries primeres (ferro, claus, pintura,...) que necessiten. Això es fa a través d'uns robots mòbils anomenats **màquines-eina** que tot voltant entre els robots fixos els porten el que aquests han de menester per poder treballar. L'ordinador també **optimitza els recorreguts** de les màquines-eina per tal que vagin d'un

lloc a l'altre sempre pel camí més curt i no quedin contra altres màquines.

- **llegeix** totes les comandes rebudes durant el dia, i en fa una **simulació de fabricació**, per trobar l'ordre en què cal treballar l'endemà, per tal de fer la feina el més ràpidament i econòmicament possible.

- porta un **control del magatzem**, de manera que minimitza els estocs i fa que sempre hi hagi una quantitat mínima preestablerta de cada peça.

- **elimina temps morts** entre feina i feina i així aconsegueix que un robot no hagi d'esperar que un altre acabi.

- **reprograma** els robots quan s'ha de canviar el tipus de peça a realitzar.

- **controla els moviments de les peces** d'una unitat a l'altra a través d'uns robots mòbils anomenats **carretons** i optimitza el seu recorregut.

Tot això fa que, malgrat la inversió inicial en equipament informàtic, les peces fabricades en un taller flexible surtin d'un 20 % a un 40 % més barates que les fabricades en un taller convencional.

Tots sabem la competitivitat que hi ha actualment entre les empreses i com es venen de bé en el nostre mercat els productes japonesos. Això és perquè la majoria dels seus tallers són flexibles, cosa que els permet per una banda abaixar els preus i per l'altra renovar-se constantment, treure models nous contínuament.

Per trobar tallers flexibles no cal anar tan lluny. Per exemple, les centrals de RENAULT i CITROËN ho són. Això fa que quan anem a una fira poguem veure vehicles realment estranys (els cotxes del futur), prototips pensats per a una exposició, que mai es fabricaran en sèrie per ésser comercialitzats. El que és més important és que en un taller d'aquest tipus totes les feines rutinàries i repetitives passen a mans dels robots i queda per a l'home la feina realment interessant i enriquidora: la invenció i la creació.

Estructura

d'un ordinador

Les caixes negres

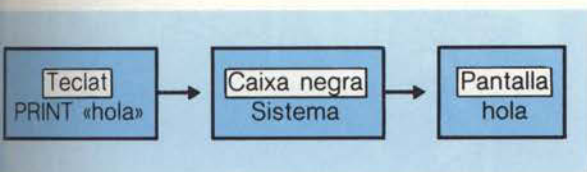
Quan al carrer parlem de «caixes negres» tots pensem en el mateix: un dispositiu, o mecanisme, que tothom sap el que fa però ningú sap com; un sistema del qual coneixem les dades d'entrada i de sortida però no sabem com s'ho manega per passar d'unes a altres.

Si en asseure'ns per primer cop davant d'un micro, sense més explicacions, ens diuen que en posar:

PRINT «Cadena»

i prémer ENTER, a la pantalla ens sortirà la cadena de caràcters que haguem escrit entre cometes, en aquest moment, tot el que hi ha entre el teclat i la pantalla serà una caixa negra.

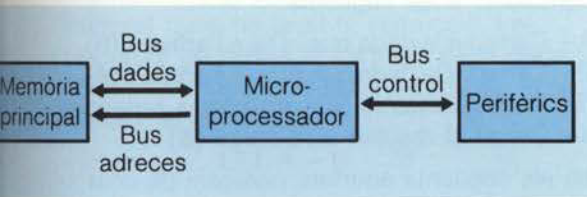




Actualment, però, d'aquesta caixa negra en sabem força coses; els seus components són: el processador (o microprocessador), la memòria central, uns interconnectadors amb els perifèrics, i els busos que connecten tots aquests elements. Aquests dispositius junt amb els perifèrics constitueixen l'estructura física que suporta la informació en un ordinador, el **hardware**.

Hem estructurat l'ordinador dividint-lo físicament i sabem les funcions que desenvolupa cadascun dels seus components. El cor d'aquesta estructura és el microprocessador, i és a través d'ell que ens anirem apropant al suport electrònic del micro. El nostre propòsit és arribar a descobrir quines eines usa per poder processar els senyals elèctrics.

Comencem parlant de l'estructura externa del microprocessador. El seu entorn està format per busos de tres tipus diferents, la memòria principal i els perifèrics.



El bus de dades és el conducte per on el microprocessador intercanvia instruccions i dades amb la memòria principal. Perquè aquest intercanvi tingui sentit el microprocessador ha de saber on és la informació que necessita, o sigui quina és l'adreça de memòria on s'ha emmagatzemat; aquesta adreça, l'envia el processador a la memòria a través del bus d'adreces. El tercer bus, el de control, s'encarrega d'enviar les ordres del processador cap a les restants unitats del micro; són ordres del tipus

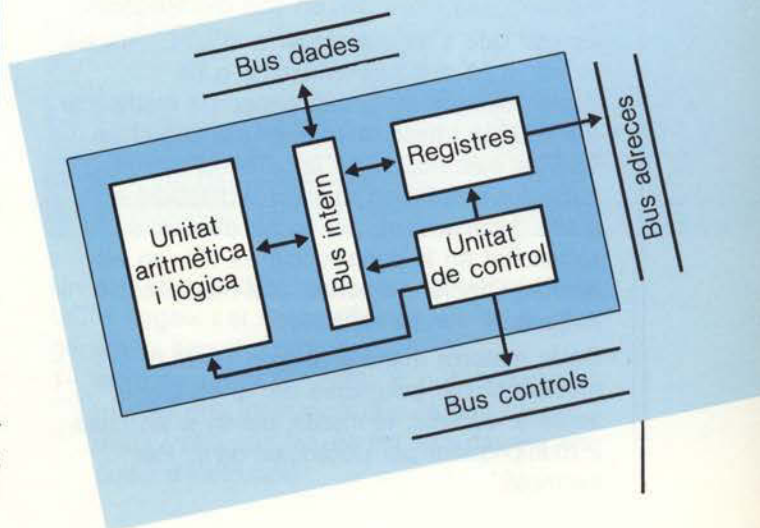
«efectuar una lectura de cinta» o bé «rebre un missatge des d'una altra unitat».

Una de les característiques més importants d'un micro és la quantitat de bits que poden transportar el bus de dades i el d'adreces. Depenent del micro, el bus de dades suporta 8 o 16 bits alhora, i 32 en els més potents. Com més gran sigui aquest nombre, més tipus de dades i d'instruccions podrà processar el microprocessador. La quantitat de bits que pot suportar el bus d'adreces està directament lligada amb la capacitat de memòria. Així, si el bus té 16 fils, un per cada bit, que és el normal en els micros, es poden adreçar $2^{16} = 65536$ posicions de memòria (bytes).

El processador

Coneixent l'estructura externa del microprocessador i quina és la seva informació d'entrada i sortida, el pas següent és intentar esbrinar què hi ha dins la «caixa negra» que el representa. Traiem la «tapadora» i veiem que està format per:

- La unitat aritmètica i lògica
- La unitat de control
- Els registres
- Un bus intern de dades
- Unes línies de control



A la unitat aritmètica i lògica s'efectuen les operacions lògiques, com ara l'AND, i les aritmètiques, totes elles en funció de l'operació bàsica de sumar. Els registres contenen l'adreça de la instrucció següent a la que s'està executant, l'adreça de les dades de la instrucció que s'està processant, les pròpies dades i el codi de la instrucció.

La unitat de control té com a tasca més important descodificar el codi de la instrucció que s'ha d'executar per tal de saber de quin tipus és, quantes dades necessita per efectuar-la, on són aquestes dades, quins dispositius s'han d'activar i, sobretot, controlar la seqüència d'accions bàsiques necessàries per executar-la.

El bus de dades intern és com els altres tres busos que hem vist abans. I les línies de control són el suport físic de les ordres que emet la unitat de control. Són fils elèctrics.

Abans de parlar més detalladament d'aquests components s'ha d'aclarir un concepte: la informació d'entrada al processador, instruccions i dades, no és exactament la mateixa que hem escrit en el programa. Hem de recordar que als micros hi ha un intèrpret (i en alguns un compilador) que transforma el nostre programa en un seguit d'ordres codificades de tal manera que siguin entenedores per a la unitat de control.

Pensem, per exemple, en una multiplicació. És clar que s'ha d'efectuar a la UAL. Ara bé, com? Hi ha dues possibilitats: o bé implementar-hi un circuit capaç de multiplicar dues xifres transformades en un seguit de senyals elèctrics, o bé aprofitar el circuit sumador que tenen tots els ordinadors a l'UAL. Normalment, per raons de cost i simplicitat en la construcció és tria, en els micros, l'opció d'efectuar una multiplicació mitjançant sumes repetides.

De la mateixa manera que en base 10 multiplicar per potències de 10 és afegir zeros a la dreta, el mateix passa si en base 2 multipliquem per potències de 2. Per exemple:



$$\begin{aligned} 26 \cdot 9 &= 11010 \cdot 1001 = 11010 \cdot (1000 + 1) = \\ &= 11010 \cdot (2^3 + 2^0) = \\ &= 11010000 + 11010 = 11101010 = 234 \end{aligned}$$

Segons això, per multiplicar n'hi ha prou amb disposar d'un sumador i d'un circuit per afegir zeros a la dreta.

Quan l'intèrpret es troba la instrucció

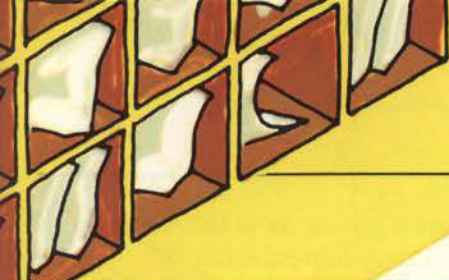
LET B=B*A

ha de «generar» un seguit d'instruccions semblants a les següents:

- 1- Carregar la dada que hi ha a l'**adreça (B)**
- 2- Carregar la dada que hi ha a l'**adreça (A)**
- 3- Multiplicar-les mitjançant sumes repetides
- 4- Deixar el resultat a l'**adreça (B)**

en els següents apartats parlarem de com són els REGISTRES i la UAL, i de quina funció exerceix la UNITAT DE CONTROL sobre cadascuna d'aquestes parts.

Notem que tot i que hem anat baixant el nivell de definició, encara usem caixes negres per explicar com funciona un micro. De totes maneres, analitzar un sistema d'aquesta manera, de forma descendent, serveix per a no perdre en cap moment una visió global de quines són les funcions de cada component del sistema, i així permet analitzar cadascuna d'elles de forma separada.



Els registres

Per executar un programa el microprocessador necessita unes posicions de memòria interna on realitzar les operacions que esdevenen necessàries un cop s'ha interpretat la instrucció. També s'hi ha de guardar informació per seguir l'execució del programa sense errors. L'esquema del bloc que hem anomenat registres és el següent:

AC	ACUMULADOR
CP	COMPTADOR DE PROGRAMA
CD	COMPTADOR DE DADES
RI	REGISTRE D'INSTRUCCIÓ

L'acumulador és el registre on es col·loca la dada sobre la qual s'ha d'operar. En el comptador hi ha l'adreça on comença la instrucció que s'ha d'executar. El registre d'instrucció té el codi que ha generat l'interpret quan ha llegit la instrucció. I el comptador de dades té l'adreça de la dada sobre la qual opera aquesta instrucció. Per exemple, si tenim

LET A = B + 20

i sigui:

C0: Codi de «carregar el contingut d'una posició de memòria»

D6: Codi de «sumar a l'acumulador un valor constant»

C1: Codi de «portar a memòria el contingut de l'acumulador»

A1: Adreça on comencen els codis generats per l'interpret

X1: Adreça de la posició de memòria B

X2: Adreça de la posició de memòria A

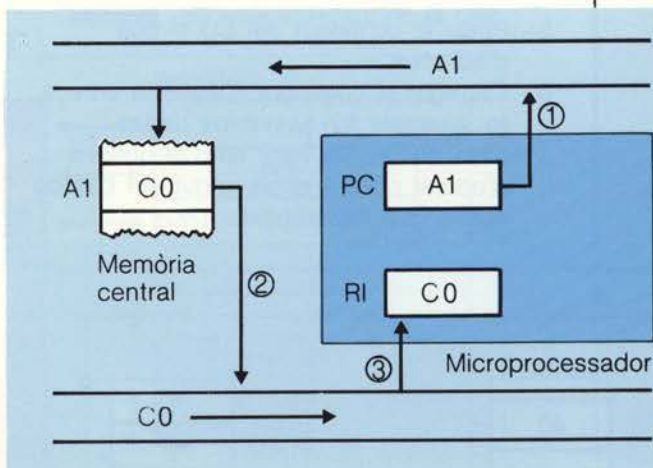
Aleshores, dins la memòria tindrem:

ADREÇA	CONTINGUT
A1	C0
A2	X1
A3	D6
A4	20
A5	C1
A6	X2

En principi, dins el CP hi haurà



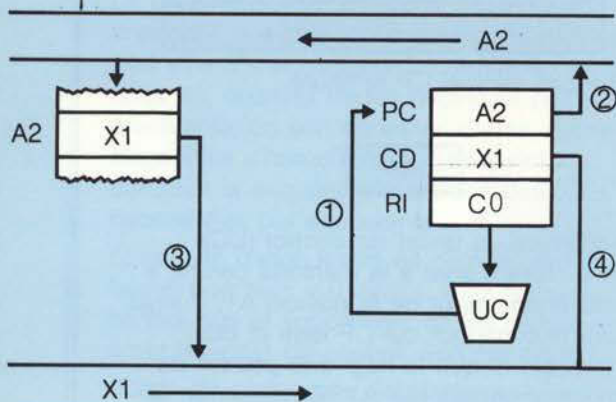
seguidament, la unitat de control (UC) donarà ordre d'anar a la memòria central a buscar el contingut de la posició A1, o sigui posarà el contingut del CP dins el bus d'adreces, i el que hi hagi a la posició de memòria apuntada per A1 ho posarà en el bus de dades; després el que hi ha en el bus de dades ho col·locarà dins el RI



Un cop el codi d'instrucció ha arribat al RI, la UC el llegeix i el descodifica. Això vol dir que genera el següent d'ordres:

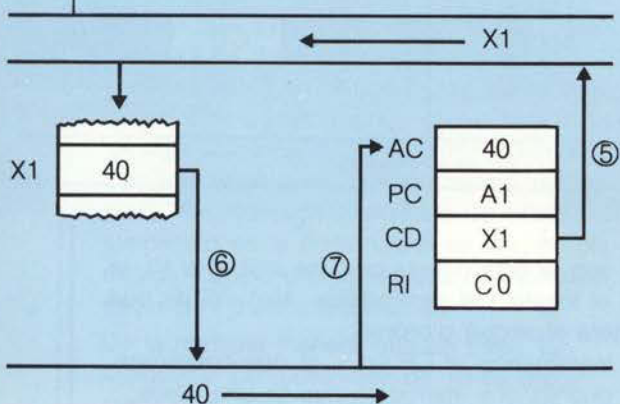
- 1- Incrementar el CP en 1, ja que comprova que és una instrucció que té una única dada i sap que les dades van darrera els codis d'instrucció

- 2- Posar el contingut del CP al bus d'adreces.
- 3- Posar el contingut d'aquesta adreça al bus de dades.
- 4- Carregar el que hi ha dins el bus de dades a DC ja que sap que és una instrucció on la dada no és un valor sinó que ho és l'adreça on està aquest.

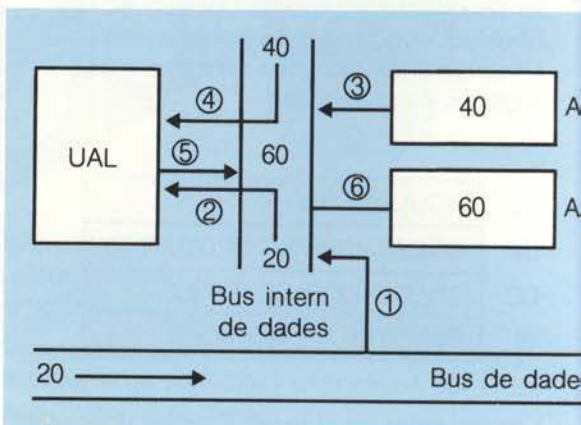


Tot seguit donarà les ordres necessàries per:

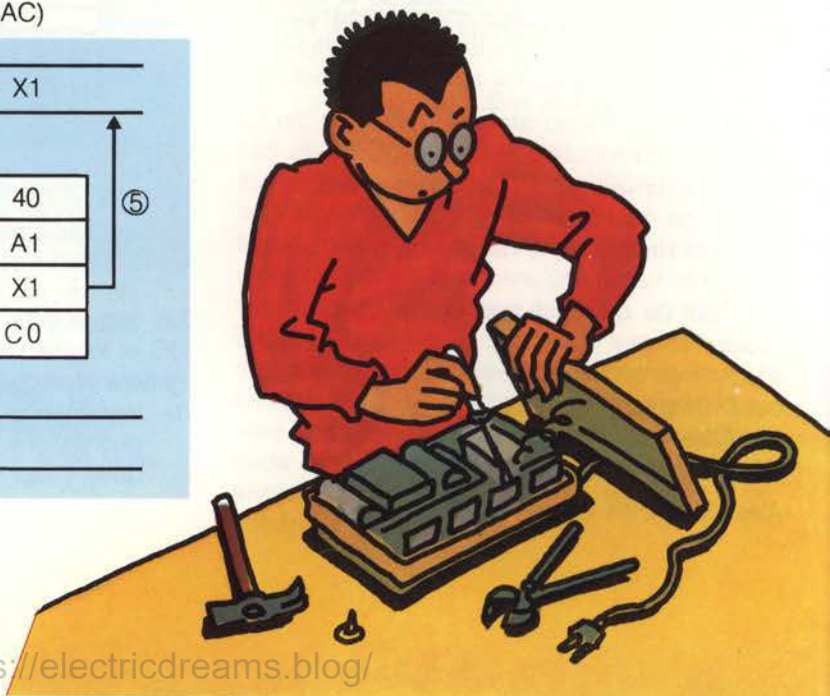
- 5- Posar el contingut del CD al bus d'adreces
- 6- Carregar el contingut d'aquesta adreça (p. exemple 40) al bus de dades.
- 7- Com que és un valor amb el qual ha d'operar posarà el contingut del bus de dades dins l'acumulador (AC)



Totes aquestes ordres de la UC han estat generades com a conseqüència de descodificar el codi d'instrucció C0. Hem de pensar però que tot el que entra dins el processador des del bus de dades ha de passar pel bus intern de dades. Un cop s'han complert les anteriors, la UC farà que el CP prengui el valor A3, i obtindrà el codi d'instrucció D6, el descodificarà i mitjançant la UAL sumarà 20 al contingut de l'acumulador AC, i deixarà el resultat en AC.



Però, què és el que passa quan les dades entren dins la UAL i la unitat de control dóna l'ordre de sumar? Per saber-ho hem de mirar que hi ha dins la UAL.



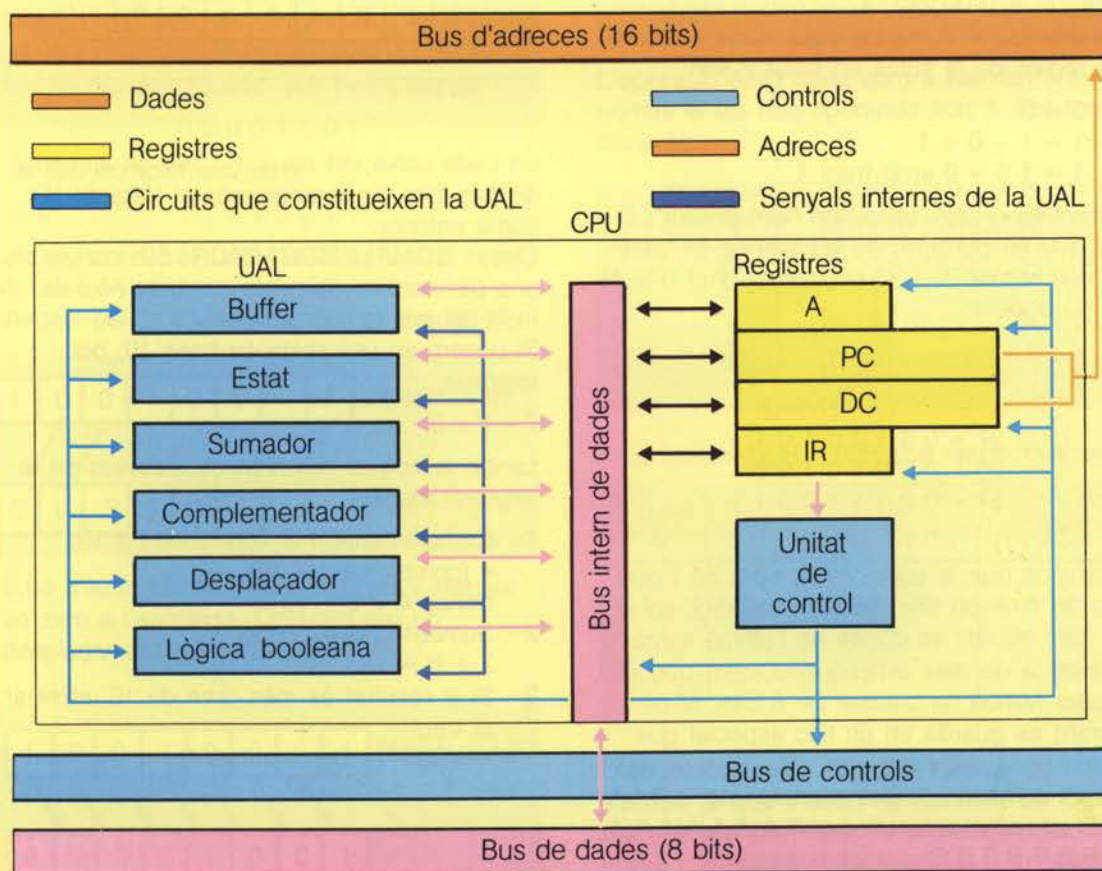


La calculadora de micro

La unitat aritmètica i lògica d'un ordinador, sigui gran, petit o mitjà, té com a propòsit fonamental efectuar:

- operacions aritmètiques bàsiques: sumes, restes,...
- operacions lògiques: AND, OR, NOT
- avaluacions de funcions numèriques: sinus, logaritmes,...

En aquest apartat estudiarem la composició bàsica d'una UAL, ja que el que ens interessa és tenir una visió, si no completa, sí detallada, del seu funcionament. En l'esquema següent tenim la totalitat de components d'un microprocessador bàsic.



Hi veiem que la UAL està formada per diversos dispositius, alguns d'ells són circuits i els altres són registres. Per tal d'explicar cadascuna d'aquestes components usarem dades representades en bytes (8 bits).

El registre **BUFFER** és un acumulador intern de la UAL per guardar-hi dades provisionalment, i permet així, que l'acumulador general AC del

microprocessador es reservi la majoria de les vegades per guardar-hi només les dades d'entrada i sortida d'un càlcul, alhora que es fan més àgils les operacions. Normalment a la UAL hi ha més d'un buffer, de la mateixa manera que també hi ha diversos acumuladors. En el nostre esquema, per executar una operació amb dos operands, en posaríem un dins l'acumulador, l'altre dins el buffer, i després els operariem.

El circuit **SUMADOR**: és el circuit electrònic que efectua la suma de dues xifres binàries. Les regles de la suma en base 2 són:

$$0 + 0 = 0$$

$$0 + 1 = 1 + 0 = 1$$

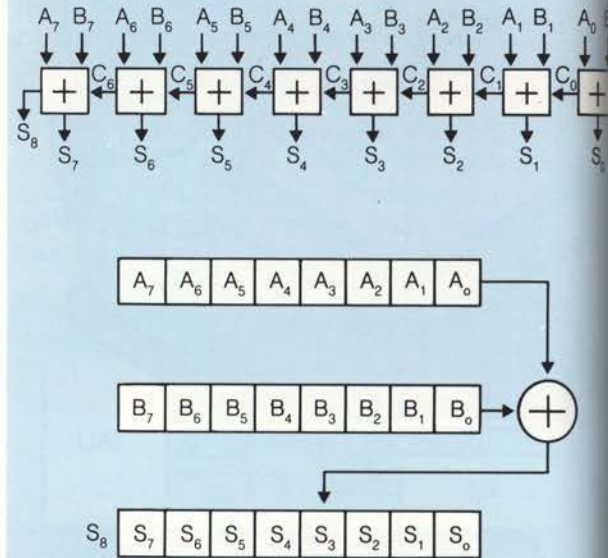
$$1 + 1 = 10 = 0 \text{ amb tragi } 1$$

El tragi és el que coneixem normalment per «les que en portem», de les sumes en base 10. Així sumar 20 i 37 en binari a vuit bits és fer el següent:

$$\begin{array}{r}
 \textcircled{1} \text{ tragi} \\
 37 = 00100101 \\
 20 = 00010100 \\
 \hline
 57 = 00111001
 \end{array}$$

Fixem-nos que la suma es fa bit a bit i que s'ha de tenir en compte si hi ha tragi, en el cas que aquest es donés en l'última suma de bits, la de més a l'esquerra, com que el resultat també ha d'ésser de 8 bits, el bit sobrant es guarda en un lloc especial que veurem més endavant.

Així un sumador de dos números de vuit bits A i B, el representarem per $A_7A_6A_5A_4A_3A_2A_1A_0$ i $B_7B_6B_5B_4B_3B_2B_1B_0$, tindrà la següent estructura:



on cada caixa del tipus \oplus és un sumador de dos bits d'operands més el tragi de la suma anterior.

Circuit **COMPLEMENTADOR**, que canvia els uns per zeros i els zeros per uns. Això és molt útil per fer les restes.

Si posem en una resta en base 10, per exemple:

$$7 - 4 = 3$$

també la podem fer mitjançant sumes de la següent manera:

- 1- Complementem el subtrahend a 10
 4 (complement a 10) = 6
 doncs $4 + 6 = 10$
- 2- Sumant-lo al minuend
 $7 + 6 = 13$
- 3- Si el resultat és més gran de 10, eliminar el tragi
 (eliminar l'1 del resultat). 3

4- Si el resultat és menor de 10, voldrà dir que en realitat la resta dona un valor negatiu, aleshores el complementem i el que ens doni serà el resultat de la resta, recordant, però, que és negatiu.

Així: $7 - 9 = -2$

el complement de 9 és 1 ;

$7 + 1 = 8 < 10$

i el complement de 8 és 2

En la base binària les restes es fan també mitjançant sumes i complementacions; aquestes últimes es fan:

1- Canviant els 1 per 0 i els 0 per 1 en el subtrahend

2- Sumant 1 al nou número binari

un cop complementat el subtrahend se suma al minuend.

Recordem que aquests números binaris són dades i, per tant, de longitud fixa, que ve donada per la capacitat del bus de dades.

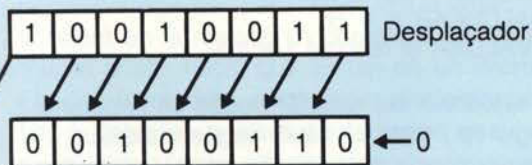
Així, el número 13 amb vuit bits s'escriu:

0 0 0 0 1 1 0 1

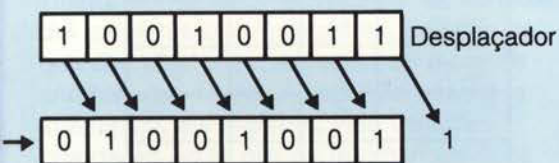
i el seu complement serà:

1 1 1 1 0 0 1 1

Un registre **DESPLAÇADOR**, que treu el bit de més a l'esquerra del número binari i afegeix per la dreta un zero. Per exemple:



o bé treu el bit de més a la dreta i afegeix un zero a l'esquerra. Obtenim així un desplaçament a la dreta com ara:



En realitat el desplaçador és una barreja de registre i circuit, ja que per un costat pot emmagatzemar informació i per l'altre pot transformar-la.

Els circuits de **lògica booleana** permeten a la UAL d'efectuar operacions lògiques com l'AND, l'OR, el NOT i la XOR. Aquestes operacions s'efectuen bit a bit sobre el contingut d'un registre, en el cas del NOT, o bé sobre el contingut de dos registres en les altres tres. L'operació NOT, el que fa és canviar els uns per zeros i els zeros per uns. Així, per representar-ho en una taula, escriurem:

A	NOT(A)
0	1
1	0

L'operació AND dona com a resultat 1 si i només si els dos operands són 1. Per tant la taula és:

A	B	(A) AND (B)
0	0	0
0	1	0
1	0	0
1	1	1

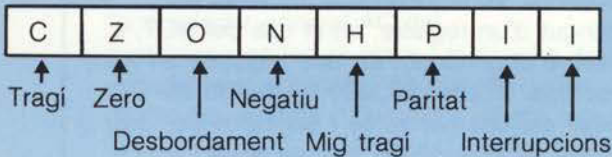
i les taules d'OR i XOR són respectivament:

A	B	(A) OR (B)	(A) XOR (B)
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

D'aquestes taules se'n diu **Taules de Veritat**, ja que s'usen normalment en lògica per representar relacions entre afirmacions falses i veritables, assignant un 0 al fals i un 1 al cert.

Els circuits que calculen las operacions AND, OR, NOT i XOR es diuen **portes lògiques** i les veurem al següent apartat.

El registre d'**ESTAT** és un registre en el qual cada bit que el conforma té un significat especial. Veiem el seu esquema:



que correspon al d'un microprocessador de vuit bits. El bit C o bit de tragi pren el valor 1 quan l'última operació ha donat un tragi 1; altrament, pren 0. Això pot passar, per exemple, quan, en el sumador, la suma de A_7 , B_7 i C_6 dona 10 o 11; aleshores el valor de S_8 s'emmagatzema en el bit C. També pren el valor 1 en els desplaçaments quan surt un 1.

El bit Z o el bit zero, pren el valor 0 quan el resultat de l'última operació és zero, i val 1 en el cas contrari. Així, si suméssim $20 + (-20)$, en l'acumulador hi hauria un 0 representat en vuit bits i en el bit Z hi hauria un 0.

Abans de continuar amb la resta de bits del registre d'estat cal dir que, normalment, el bit de més a l'esquerra de la representació d'una dada s'usa com a bit de signe. Hi ha, però, diverses maneres de representar un número. Segons aquesta regla (la més entenedora) s'usa el primer bit per al signe i els set restants representen el valor absolut de la quantitat. Se'n diu notació **signe-magnitud**. Així, 60 i -60 es representarien:

60 =	0	0111100
-60 =	1	0111100

El bit 0 o bit de desbordament es posa a 1 quan el resultat d'una operació no es pot

representar en els bits que té assignats. Per exemple, en vuit bits i notació signe-magnitud, es poden representar números entre:

$$11111111 = -127 \quad \text{i} \quad 01111111 = 127$$

Si volem sumar, per exemple, 80 i 60 (=140) amb vuit bits ens passarà el següent:

80 =	0	1	1	0	1	0	0	0	0
60 =	0	0	1	1	1	1	0	0	0
		1	0	0	0	1	1	0	0
									= -12

i dona un resultat incorrecte, ja que la suma de dos números positius (bit de signe = 0) no pot donar un negatiu (bit de signe = 1). El bit N, o bit de resultat negatiu, es posa a 1 quan el resultat de l'última operació ha estat negatiu (menor de zero).

Els bits Z i N són molt útils per avaluar condicions sobre valors numèrics del programa escrit en BASIC. Per comparar dos valors, el programa en llenguatge màquina que sempre es genera té la següent estructura:

BASIC: IF A # B THEN...

LLENGUATGE MÀQUINA:

CALCULAR A - B

AVALUAR el resultat i actuar.

L'avaluació del resultat depèn de l'operand lògic (#) usat. La taula següent explica l'equivalència entre l'operand # i els valors de Z i N després de calcular A-B

#	Z		N
<	1	i	1
<=	0	o	1
=	0	i	0
>=	0	o	0
>	1	i	0

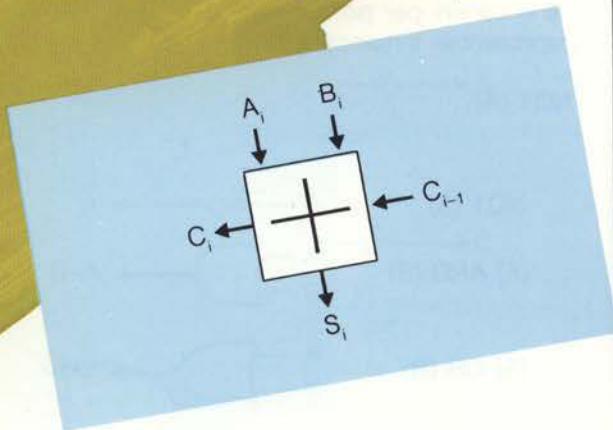


El bit H o de mig traçí es posa a 1 quan, en una operació sobre operands de vuit bits, el resultat d'operar els quatre primers de la dreta dona un traçí d'1.

El bit P ó de paritat serveix per controlar, d'una manera senzilla, els errors d'emmagatzematge d'informació després d'una transmissió d'aquesta entre el microprocessador i els dispositius externs com ara cintes, memòries, etc... Es posa a 1 si el número de bits a 1 de la dada és senar i a zero quan és parell. No és un mètode gaire efectiu per detectar errors (i, doncs, poder corregir-los) però acostuma a ésser suficient en la majoria d'operacions de trasllat d'informació que es fan en un micro. Els dos últims bits, anomenats d'interrupció o bits I, detecten qualsevol interrupció que pugui parar l'execució del programa en curs, com per exemple prémer BREAK, o bé que un dispositiu d'entrada pugui introduir una dada i aquesta introducció tingui prioritat sobre la tasca que estigui efectuant l'ordinador. Perquè el procés de tramitar una interrupció no afecti el valor de les variables del programa o dels registres, aquests es guarden dins unes posicions de memòria anomenadas PILA; un cop s'ha acabat la interrupció, els registres i les variables guardades en la pila es retornen als llocs originals.

Les portes

Hem arribat, descomposant cada dispositiu del micro en components més elementals, a saber com és la UAL per dins, què fan els registres, etc... Però, com s'ho fa, per exemple, un sumador per calcular el resultat de la suma del que hi ha dins l'acumulador i dins el buffer? Sabem que l'estructura està basada en vuit caixes negres del tipus:



Què hi ha dins aquestes caixes?

Fa una estona hem parlat de les operacions AND, NOT, OR i XOR. Aquestes funcions operaven bit a bit i les podíem representar mitjançant taules de veritat, on els uns i els zeros representaven els dos tipus de senyals elèctrics que transporten la informació. El que és realment interessant d'aquests operands és que mitjançant només dos d'ells AND i NOT, o bé OR i NOT, es pot representar qualsevol tipus d'operació que actuï sobre entrades amb valor 0 o 1 i doni com a sortida un zero o un u.

Les OR i XOR, no les posem, ja que es poden representar mitjançant la NOT i l'AND:

$$(A) \text{ OR } (B) = \text{NOT} (\text{NOT}(A) \text{ AND } \text{NOT}(B))$$

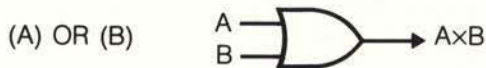
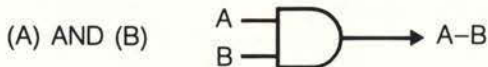
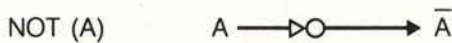
$$(A) \text{ XOR } (B) = (\text{NOT}(A) \text{ AND } (B)) \text{ OR } (\text{NOT}(B) \text{ AND } (A))$$

si representem AND amb (.), OR amb (x), NOT amb (-), i XOR amb (), ho podem comprovar mitjançant la taula de veritat:

A	B	AxB	A⊕B	\bar{A}	\bar{B}	$\bar{A}\bar{B}$	$\bar{\bar{A}}\bar{B}$	$\bar{A}\bar{B}$	$A\bar{B}$	$\bar{A}BxA\bar{B}$
0	0	0	0	1	1	1	0	0	0	0
0	1	1	1	1	0	0	1	1	0	1
1	0	1	1	0	1	0	1	0	1	1
1	1	1	0	0	0	0	1	0	0	0

Els circuits que calculen l'AND, el NOT i l'OR (aquest, l'usarem per motius de claredat) se'ls coneix per **portes lògiques**, i es representen mitjançant:

NOT (A)



Hi ha un mètode per representar les operacions amb zeros i uns (conegudes com a funcions booleanes) mitjançant aquestes tres portes.

El primer que s'ha de fer és construir la taula de veritat d'aquesta funció booleana. Això es fa posant tantes columnes com entrades tingui el circuit que la calcula, més una columna per a la sortida. En les files es posen totes les combinacions possibles que poden prendre els valors d'entrada, més la sortida que correspon a cadascuna d'aquestes combinacions. Així, en el circuit de la suma de dos bits tenim, per exemple:

$$\left. \begin{array}{l} A_1 = 1 \\ B_1 = 1 \\ C_{1-1} = 0 \end{array} \right\} \rightarrow \begin{array}{r} 1 \\ +1 \\ +0 \\ \hline 10 \end{array} \rightarrow \left\{ \begin{array}{l} S_1 = 0 \\ C_1 = 1 \end{array} \right.$$

Si fem això per a totes les tríades (A_1, B_1, C_{1-1}) , en la taula de veritat tindrem:

A_1	B_1	C_{1-1}	S_1	C_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

El següent pas a fer és, per cada columna de sortida, allà on hi hagi uns, agafar la fila de valors d'entrada que correspon:

$$S_1 : 001, 010, 100, 111$$

$$C_1 : 011, 101, 110, 111$$

Seguidament, si a la posició d'una entrada hi ha un zero, es posa el símbol que representa aquesta entrada amb el signe de negació (NOT) sobre d'ella, i si hi ha un 1 es posa únicament el símbol:

$$S_1 : \bar{A}_1\bar{B}_1C_{1-1}, A_1B_1C_{1-1}, \bar{A}_1B_1\bar{C}_{1-1}, A_1\bar{B}_1\bar{C}_{1-1}$$

$$C_1 : \bar{A}_1B_1C_{1-1}, A_1\bar{B}_1C_{1-1}, A_1B_1\bar{C}_{1-1}, A_1B_1C_{1-1}$$

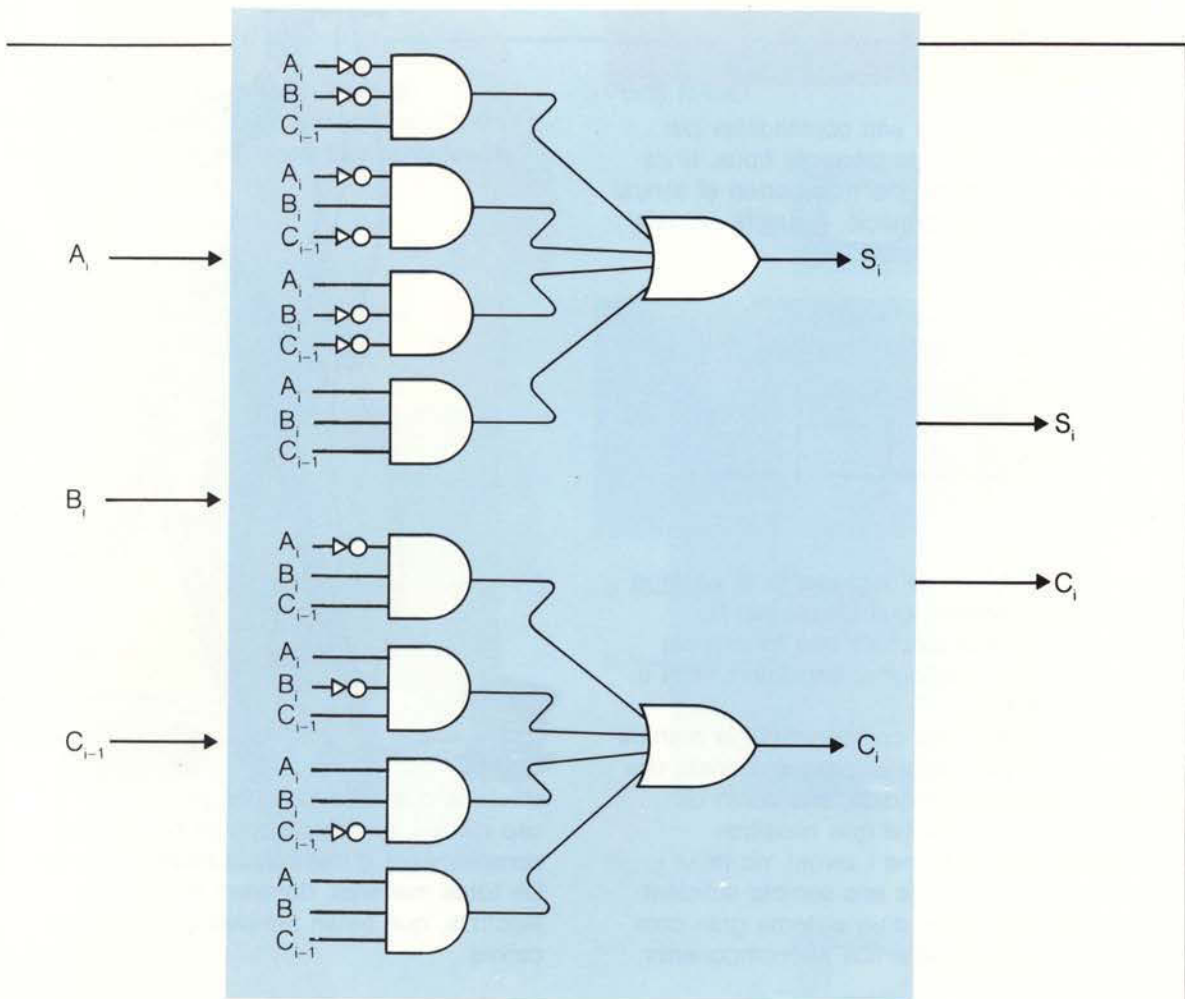
Per a cadascuna d'aquestes tríades es fa un AND entre tots els símbols i després per a cada funció de sortida es calcula l'OR entre totes les tríades

$$S_1 : A_1\bar{B}_1C_{1-1} \times A_1B_1\bar{C}_{1-1} \times \bar{A}_1B_1C_{1-1} \times A_1\bar{B}_1\bar{C}_{1-1}$$

$$A_1\bar{B}_1C_{1-1}$$

$$C_1 : A_1\bar{B}_1C_{1-1} \times A_1B_1\bar{C}_{1-1} \times A_1B_1C_{1-1} \times A_1BC_{1-1}$$

i el circuit corresponent a la caixa que representava un dels components d'un sumador fet amb portes lògiques serà:



En aquest circuit hem usat portes AND amb tres entrades i portes OR amb dues entrades; això no canvia la definició donada de l'operació AND i de l'OR, si pensem en la generalització següent:

- Una porta AND té sortida 1 si i només si totes les seves entrades valen 1, i té sortida 0 en tot altre cas.
- Una porta OR té sortida 0 si i només si totes les seves entrades valen 0, i té sortida 1 en tot altre cas.

Si en el circuit d'abans anem provant els valors d'entrada que tenim a la taula de veritat, comprovarem que si apliquem les dues anteriors i la definició de NOT, ens

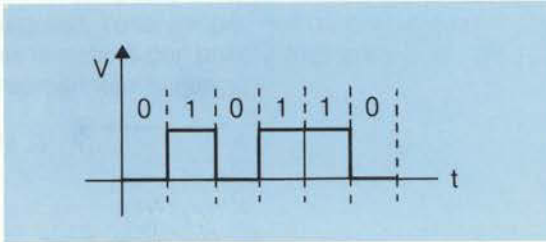
dóna a la sortida del circuit el mateix valor que tenim a la taula.

Com que tots els circuits que hem representat en l'esquema detallat del microprocessador tenen com a entrades i sortides només dos tipus de senyals elèctrics, que representem amb 0 i 1, és fàcil deduir que, sigui quina sigui la funció que efectuem, aquesta es podrà representar amb un circuit de portes lògiques AND, OR i NOT.

Tot i havent arribat al nivell de portes lògiques, de les quals sabem que les seves entrades són senyals elèctrics transportats pels fils que les representen, i l'operació que calculen, no sabem encara quins són els components elèctrics que «uneixen» aquests fils d'entrada amb els fils de sortida.

Els circuits

Les portes lògiques són constituïdes per elements elèctrics de diferents tipus, units per fils conductors que transporten el senyal que suporta la informació. Aquests senyals són digitals, de la forma

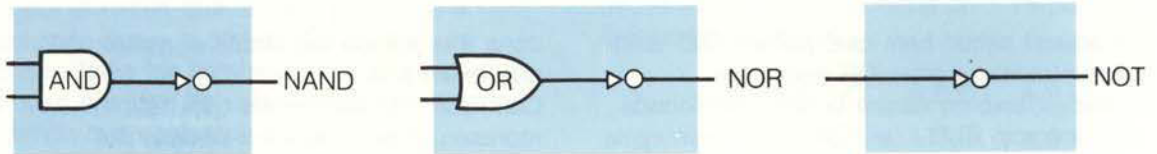


on l'altura del senyal representa el voltatge del corrent elèctric que circula pel fil. Els components elèctrics que formen els circuits són resistències, transistors NPN o PNP, díodes...

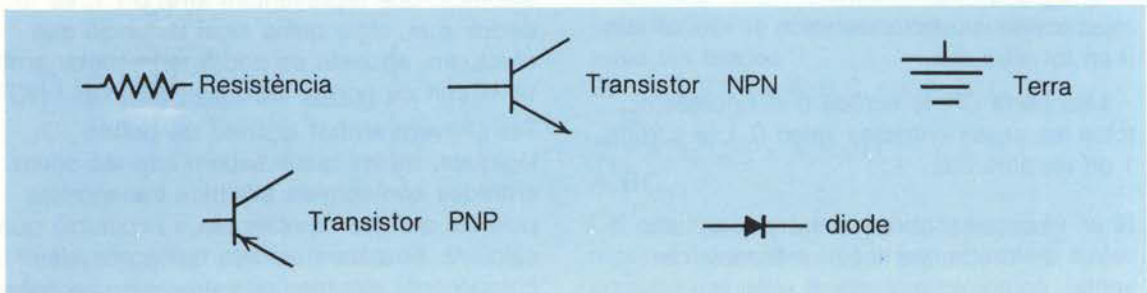
El que són aquests components i la manera com s'han de combinar perquè, donats uns certs voltatges d'entrada, ens donin un voltatge de sortida (el que nosaltres representem amb uns i zeros), no ho explicarem perquè ja ens sembla suficient mostrar que el pas d'un sistema gran com és un microordinador fins als components



elèctrics que el formen (les portes) no és cap misteri, a més perquè es necessiten coneixements d'electrònica bastant elevats. De totes maneres, donarem els circuits elèctrics, que estan representats amb les caixes

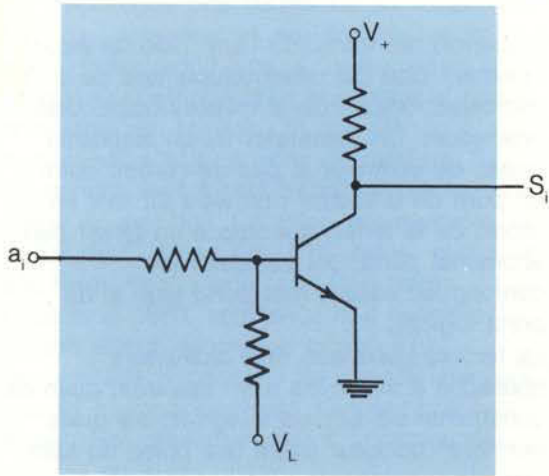
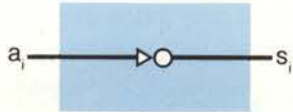


Representarem els components de la següent forma:

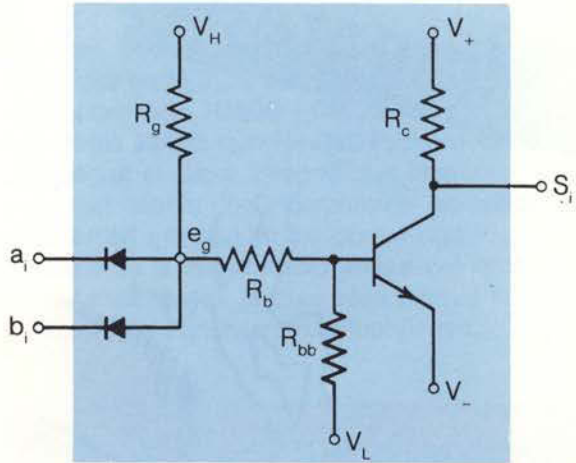
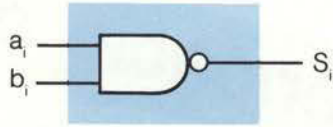


i representarem amb V_+ el voltatge corresponent a 1, V_- el corresponent a 0 i amb V_H i V_L dos voltatges que compleixen $V_H > V_+ > V_- > V_L$

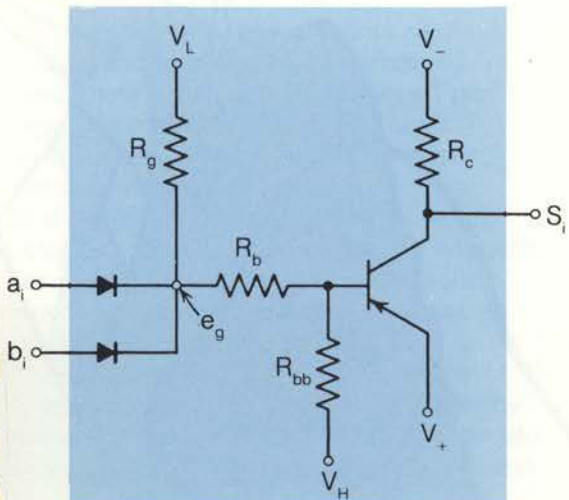
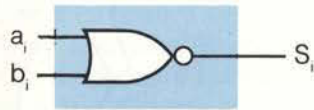
Porta NOT



Porta NAND



Porta NOR



Disseny i fabricació de circuits integrats.

La VLSI i la tecnologia MOS

L'aparició del transistor l'any 1956 ha estat l'element clau del desenvolupament de la microelectrònica i de la miniaturització dels ordinadors. Un **transistor** és un dispositiu capaç de *controlar* el pas de corrent entre un punt dit *drenador* i un altre dit *font* en funció de la tensió aplicada a un tercer punt, anomenat *porta*, del transistor (no confongueu aquest mot *porta* amb el de *porta lògica*).

La tercera generació dels ordinadors aparegué a mitjan els anys seixanta, quan es construïren els circuits integrats, els quals permeten col·locar sobre una oblea de silici, tots els components i les connexions necessàries per fer un circuit, en lloc de fabricar transistors aïllats i després connectar-los.

A partir d'aquesta època existirà sempre una interdependència entre ordinadors i tecnologia microelectrònica.

La convergència d'una tecnologia senzilla, d'unes eines de CAD potents, d'una metodologia de disseny estructurat i d'un abaratament progressiu del càlcul fa que la concepció dels circuits integrats sigui més assequible. Una de les formes actuals de disseny destinada a la concepció de circuits integrats a gran escala és la coneguda pel nom de VLSI (Very Large Scale Integration) que féu la seva aparició l'any 1978.

En VLSI es té en compte, primer, la tecnologia de circuits integrats amb una **densitat d'integració** a partir de 100.000 transistors en un sol chip (Hewlett-Packard ha aconseguit ficar-ne 450.000 en una pastilla de silici de 6.35 x 6.35 mil·límetres, i augura



que el 1986 serà capaç de ficar-n'hi uns 800.000) i, segon, les **eines** per a la concepció i el disseny de components electrònics en general, i per a la informàtica en particular.

Per dissenyar un circuit integrat d'uns 100.000 transistors sobre una oblea de silici, en VLSI són necessaris, per terme mitjà, 120 dissenyadors/any. En un futur immediat, a uns cinc anys vista, hom pensa que es podran integrar uns 10 milions de transistors per chip. Aleshores el seu disseny requerirà 6.000 dissenyadors/any. Aquestes xifres indiquen l'existència d'un buit entre el que la tecnologia pot produir i el que els dissenyadors són capaços de realitzar. En els darrers anys s'està treballant per intentar omplir aquest buit. Cal citar, per exemple, el treball de C. Mead i de L. Conway, que en el seu llibre *Introduction to VLSI systems*, publicat el 1980, aporten com a possible solució el que anomenen metodologia estructurada de disseny de circuits integrats, que ve a omplir una part d'aquest buit. Aquí volem presentar alguns aspectes del disseny en VLSI.

Aquest disseny utilitza, fonamentalment, tecnologies que operen sobre oblees de silici, o altres semiconductors, com a base de la integració dels circuits. Això no obstant, comencen a sorgir altres tecnologies basades en «arseniür de gal·li» i en les anomenades «bombolles magnètiques».

Quant a la tecnologia de silici, existeixen dues grans branques: la **bipolar** i la **MOS** (Metal Oxid Semiconductor).

En la bipolar l'efecte transistor està lligat a les dimensions verticals del chip (gruix de la pastilla de base) i es produeix en el volum del semiconductor, mentre que en la tecnologia MOS l'efecte transistor està lligat a les dimensions horitzontals i es produeix en la superfície del semiconductor.

Amb tecnologia MOS es construeixen, entre d'altres, circuits integrats digitals, transistors de potència, sensors químics i circuits analògics, encara que aquests darrers se solen fer més amb tecnologia bipolar.

La tecnologia que ha impulsat a la VLSI és la MOS, car:

- produeix transistors autoalineats i susceptibles d'ésser miniaturitzats i permet una alta capacitat d'integració
- hi ha poca dissipació de potència
- ofereix la possibilitat d'emmagatzemar càrreges elèctriques
- el procés tecnològic és relativament senzill

De totes les tecnologies MOS, les més emprades en la VLSI són precisament les conegudes com **NMOS** i **CMOS** (és amb tecnologia NMOS que Hewlett-Packard ha aconseguit el circuit integrat que hem esmentat abans) que incorporen el **polisilici** (silici amorf i dopat) en les portes dels transistors i la **implantació iònica** pel control de la tensió dintell, la qual cosa permet la utilització de transistors d'em pobriment.

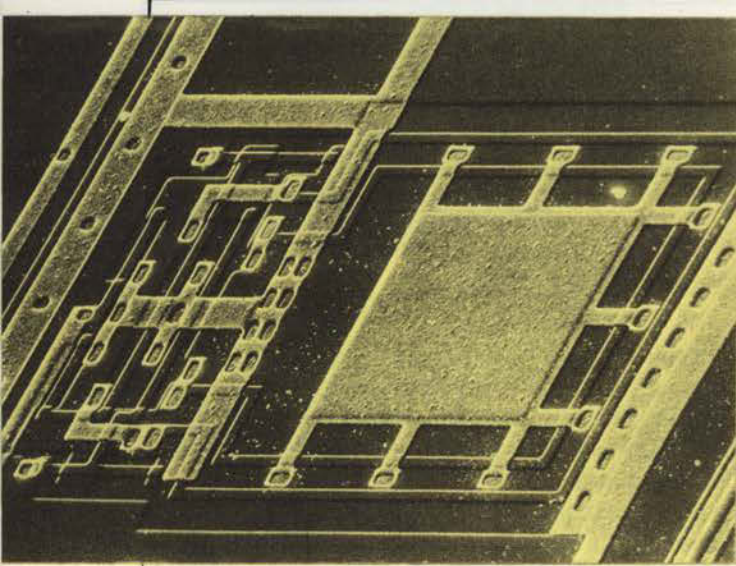
Regles de disseny d'un circuit integrat

33

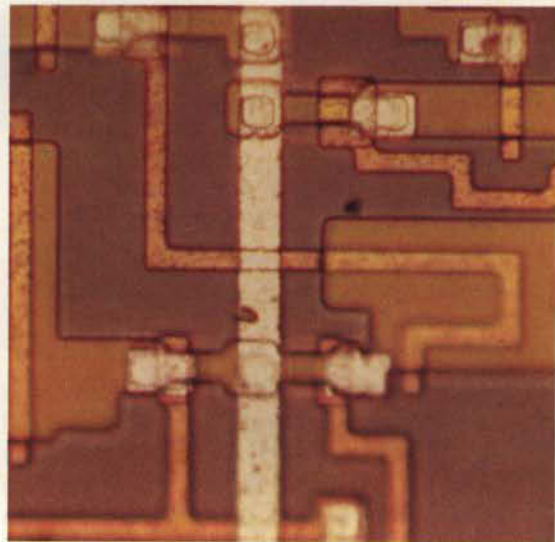
Abans hem assenyalat com un dels aspectes que fan més assequible el disseny d'un circuit integrat és la relativa simplicitat de la tecnologia MOS. Una cosa és el procés tecnològic que ha de sofrir l'oblea de silici per convertir-se en un circuit integrat (procés eminentment químic) i una altra, el disseny de les geometries dels diferents nivells d'integració formats pels materials conductors separats alternativament per capes de materials aïllants, els quals constitueixen els transistors i les seves interconnexions.

El disseny d'aquestes geometries és l'essència del disseny dels circuits integrats, car permet la definició de les diferents màscares necessàries per dur a terme el procés tecnològic de fabricació.

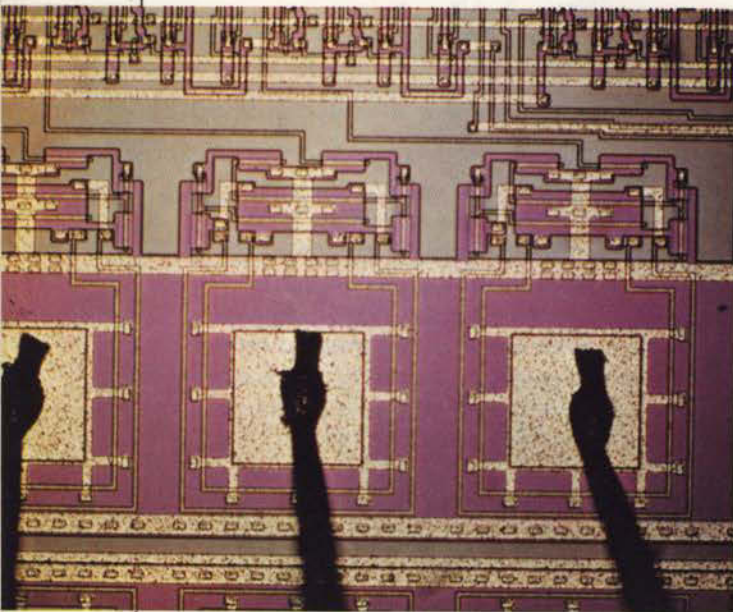
Aquest disseny és fortament simplificat gràcies a l'existència de regles de disseny tecnològic que limiten el valor de l'extensió, separació, amplada, etc., entre els diferents nivells de material.



Màscara en la fabricació de circuits integrats



Detall d'un circuit integrat amb tecnologia NMOS

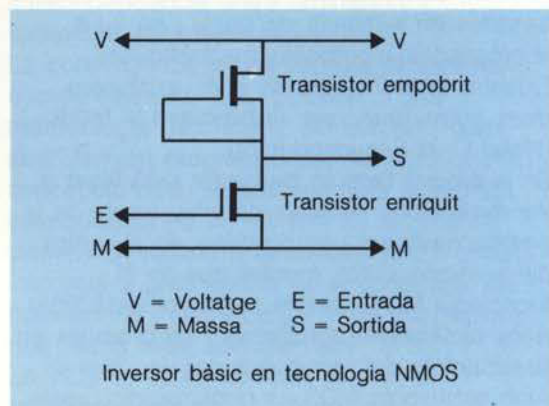


Detall d'un circuit integrat

Per fixar-nos les idees considerem que utilitzem una tecnologia NMOS, amb portes de polisilici i implantació iònica, per fer el disseny d'un circuit molt senzill: **un inversor bàsic**. En aquesta tecnologia disposem de quatre nivells: metàl·lic, polisilícic, de difusió i

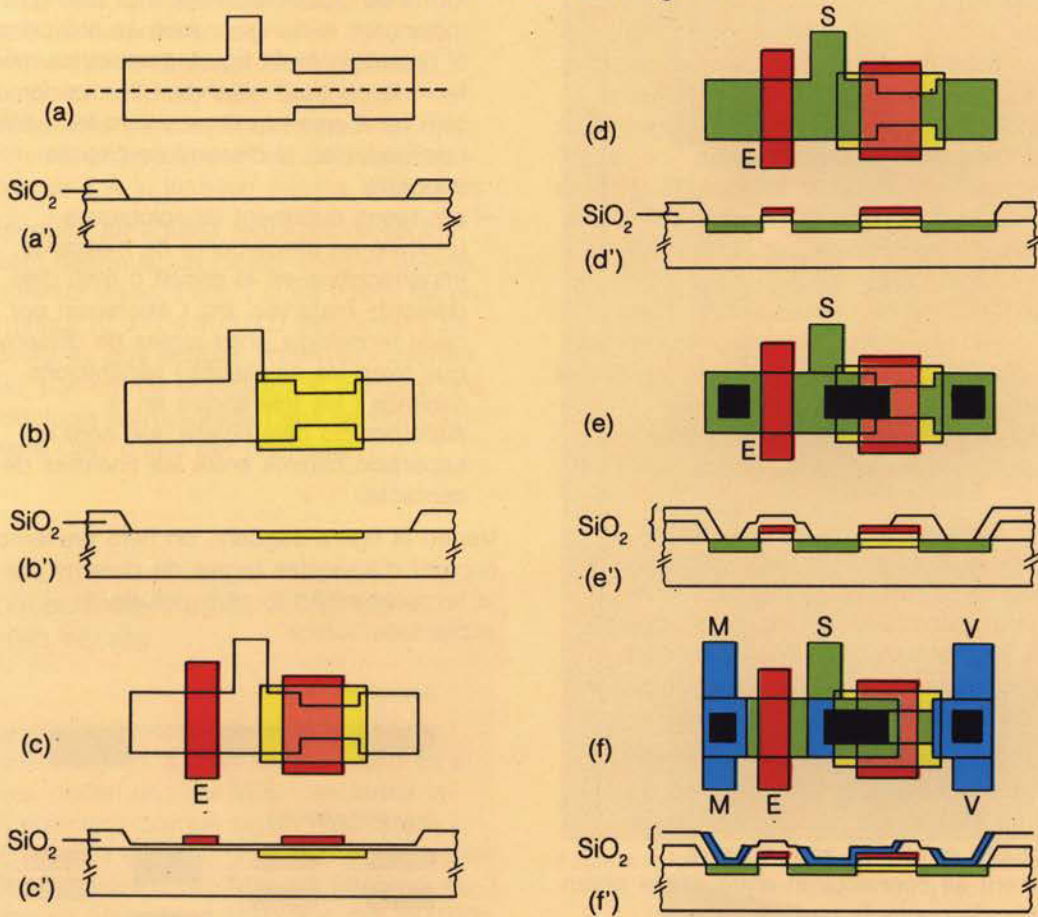
d'implantació iònica, i de dos tipus de transistors: els enriquits i els empobrits. Un transistor empobrit difereix de l'enriquit perquè en el seu canal té impureses implantades iònicament.

L'inversor que considerarem serà constituït per un transistor enriquit amb una càrrega activa produïda per un transistor empobrit amb connexió de la seva porta a la seva font (vegeu figura següent).



Per fer aquest disseny seguirem els passos següents, que trobeu representats en aquesta figura:

Processos bàsics en tecnologia NMOS



Primer: S'ha d'oxidar tota la superfície de l'oblea, i s'obtéindrà òxid de silici (SiO_2), on s'obrirà la finestra en la qual s'instal·laran els dos transistors (vegeu **a** i **a'** en la figura, on **a'** és la secció transversal d'**a**)

Segon: S'han d'introduir impureses en la regió del canal de transistor d'empobriment, mitjançant implantació iònica (zones marcades amb groc en les figures **b** i **b'**)

Tercer: S'ha de dipositar el polisilici, el qual formarà les portes dels dos transistors i les

connexions entre ells (zona marcada amb vermell a **c** i **c'**). El polisilici ha estat dipositat sobre una fina capa d'òxid de silici (SiO_2) en la regió de les portes sobre un òxid més gruixut en les altres regions, en què únicament actua com a línia d'interconnexió (entrada de l'inversor). Per tant, està aïllat del substrate de silici (vegeu **c'** de la figura).

Quart: S'ha d'introduir impureses mitjançant la difusió en tota la finestra definida en **a** menys en les zones cobertes de polisilici.

D'aquesta manera es creen els drenadors i les fonts dels dos transistors, que estaran alineats amb les portes (zones marcades amb verd en **d** i **d'**). Observem que la font del transistor d'empobriment i el drenador d'enriquiment són una sola regió (vegeu **d** en la figura) i en conseqüència estan automàticament connectats. Aquesta regió es pot perllongar (sortida de l'inversor) formant una línia d'interconnexió.

Cinquè: S'ha de fer una deposició d'una capa d'òxid gruixuda i després s'han d'obrir les finestres (veure **e'**) per efectuar contactes metàl·lics (zones negres en la figura) dipositant-hi alumini mitjançant evaporacions en el buit (zones marcades de blau en **f** i **f'**).

Aquest exemple tan senzill ens mostra l'existència de quatre nivells: implantacions iòniques (groc), polisilici (vermell), difusió (verd) i metall (blau), mitjançant els quals podem formar transistors i interconnexions. Els punts essencials de tot aquest procés, a nivell de disseny, són els següents:

Primer: Els nivells de vermell, verd i blau serveixen per efectuar interconnexions i estan autoaïllats. Els podem entrecreuar, exceptuant el vermell amb el verd, i únicament es connectaran entre ells si obrim finestres de contacte (regions negres)

Segon: Mitjançant la superposició del groc, el vermell i el verd, es crea un transistor d'empobriment.

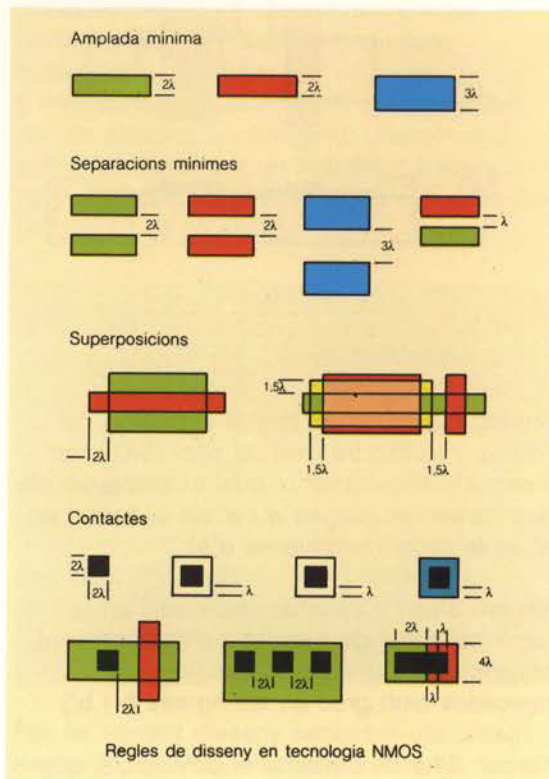
Tercer: Mitjançant la superposició del vermell i el verd es creen transistors d'enriquiment.

Amb els tres punts anteriors podem adonar-nos de la forma de dissenyar transistors i d'interconnectar-los i, en conseqüència, de com dissenyar circuits integrats.

Amb tot, la pregunta que ara sorgeix és: Com es determinen la forma i les dimensions dels diferents nivells d'un circuit integrat? La resposta ens ve donada per les tres raons següents:

- observar, en la figura precedent, que la forma de qualsevol nivell s'ha obtingut mitjançant rectangles. Això és així perquè el rectangle és la figura geomètrica més fàcil de generar mitjançant l'ordinador, i això és el que nosaltres volem fer, utilitzar l'ordinador en el disseny de circuits integrats.
- per raons purament tecnològiques (precisió en alineaments de màscares, imperfeccions en el gravat o atac dels diferents materials, etc.) existeixen per a cada tecnologia unes regles de disseny que fixen les amplades i separacions mínimes i les toleràncies en la superposició dels nivells, així com la separació mínima entre les finestres de contacte.

Vegeu la figura següent, on hem presentat el conjunt d'aquestes regles de disseny per a la tecnologia NMOS amb polisilici i implantació iònica.



- la relació entre la longitud i l'amplada del canal dels dos transistors que hem realitzat determinen una sèrie de propietats de l'inversor: tensió dintell, temps de resposta, etc. En funció d'aquestes característiques podem determinar les dimensions dels canals en fixar en 2λ la longitud mínima d'un canal.

Respectant tot aquest reguitzell de regles haurem dissenyat un circuit integrat que és un inversor bàsic. Però, i si en volem fer un que tingui 10.000 o 100.000 transistors? La resposta és que seguint aquestes mateixes regles podrem fer el disseny si utilitzem una metodologia estructurada i ens servim d'un sistema CAD d'ajut per ordinador.

Fabricació de circuits integrats a gran escala

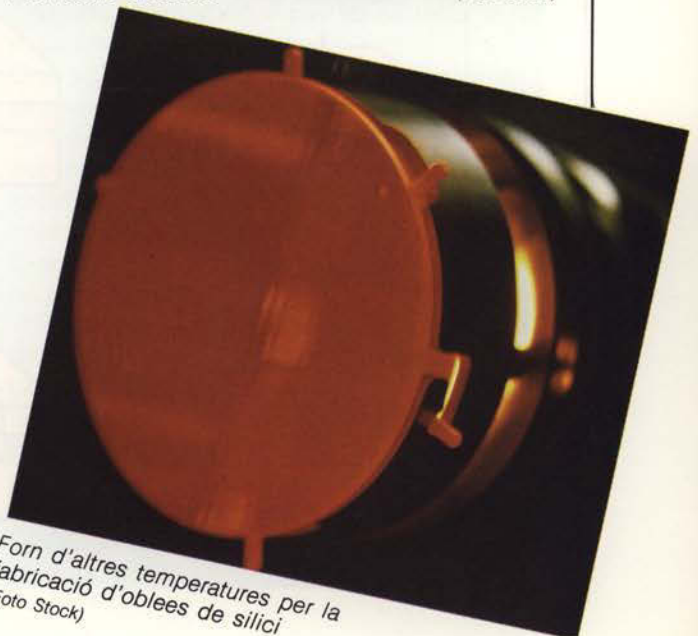
L'objectiu fonamental de la fabricació de circuits integrats a gran escala, i això és el que es pertén en usar VLSI, és reduir la seva grandària perquè siguin més ràpids i gastin menys energia, cosa que minimitza els costos. El disseny dels circuits integrats es realitza en un nombre al voltant del 100.000 transistors per chip. Aquest disseny es porta a terme amb ajut de programes CAD, a una escala acceptable per a la seva visualització. La fabricació dels circuits integrats es fa en tres dimensions, mitjançant capes diferents que se'n diuen **màscares**. Algunes de les capes estan a l'interior i altres a la superfície de l'oblea de silici. El procés de fabricació consisteix en la superposició de les màscares tot seguint l'ordre assignat pel dissenyador.

Un cop establertes les especificacions funcionals del circuit i dels processos necessaris per a la seva fabricació, es realitza el disseny real, generalment amb l'ajut d'una pantalla gràfica connectada a un ordinador, en la qual es representa el circuit.



Fabricació d'obleses

(Foto Stock)



Forn d'altres temperatures per la fabricació d'obleses de silici

(Foto Stock)

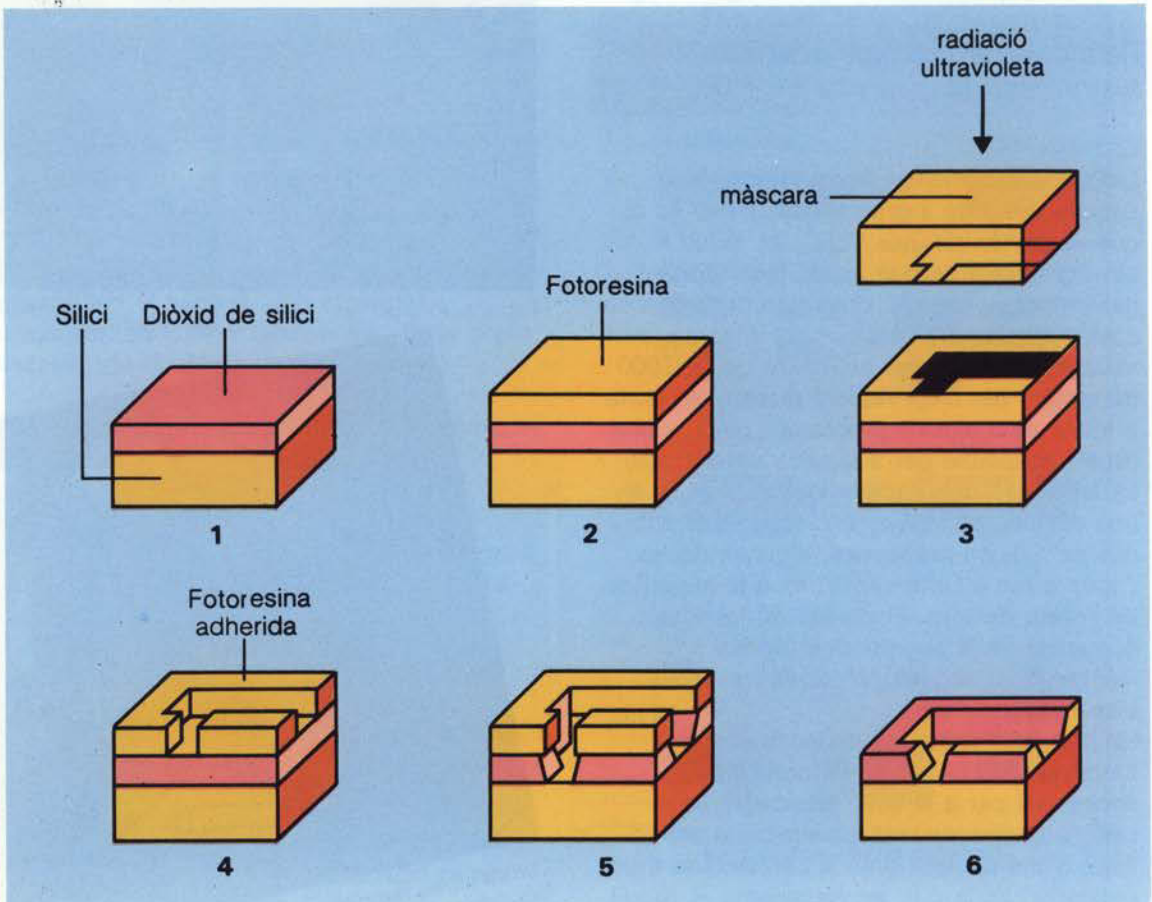
A partir del disseny es fabricaran les màscares de cada una de les capes del circuit mitjançant un procediment fotogràfic. Aquestes màscares són ja a grandària real del circuit i es componen de zones transparents i zones obscures que després s'utilitzaran per realitzar el procés de fabricació pròpiament dit.

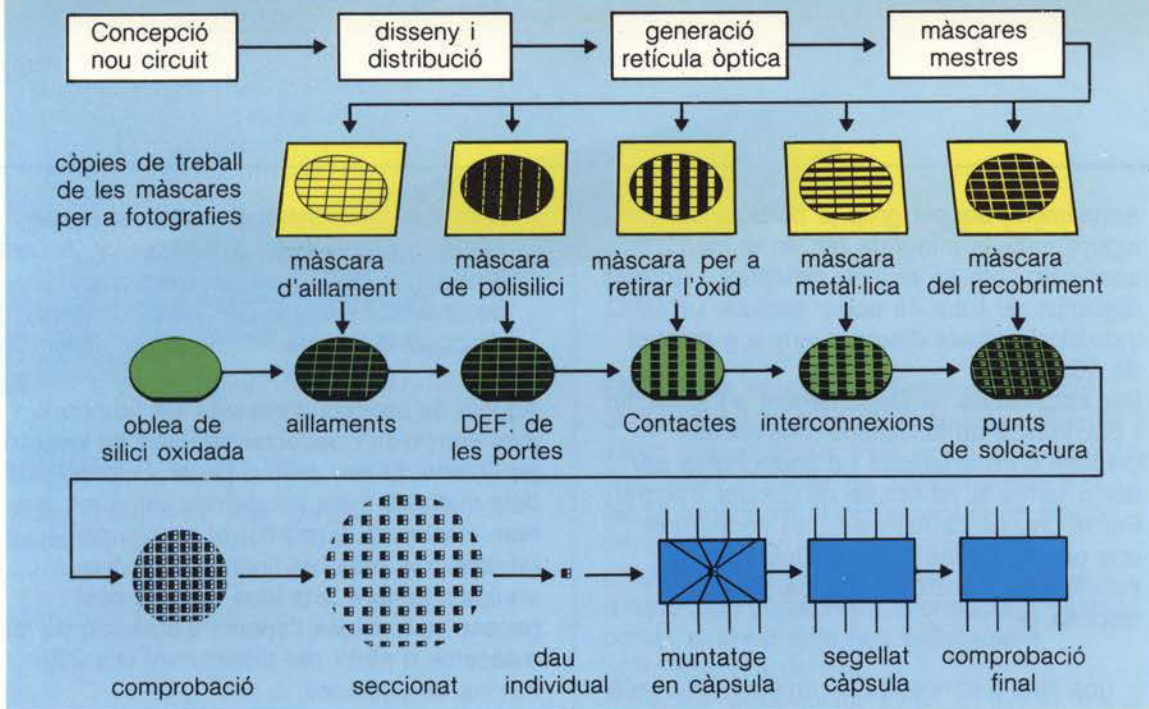
Per passar de les màscares al circuit integrat s'utilitza un procés de fotogravació dit **fotolitografia**. Sobre una oblea de silici amb una capa superior d'òxid de silici es col·loca una gota de fotoresina diluïda amb dissolvent. A continuació es fa girar ràpidament l'oblea, mentre que el dissolvent

s'evapora i només resta la fotoresina, i s'aconsegueix una pel·licula fotosensible sobre la superfície. Seguidament s'exposa a la llum ultravioleta a través de la màscara, amb la qual cosa la fotoresina es fa insoluble a una solució reveladora i així es pot retirar de les parts en què la màscara era obscura. Després se submergeix l'oblea en una solució d'àcid fluorhídric que ataca selectivament l'òxid de silici i manté inalterats el disseny de la fotoresina i el substrat de silici. Finalment, mitjançant un altre tractament químic, es treuen les restes de fotoresina, amb la qual cosa la capa queda acabada.

Il·lustració gràfica de les sis etapes de què consta un procés fotolitogràfic:

1. Oblea Oxidada. **2. Recobriments amb fotoresina.** **3. Exposició a la llum ultravioleta.** **4. Diòxid de silici atacat amb àcid fluorhídric.** **5. Retirada del patró de fotoresina.** **6. Resultat del tractament químic final.**





Resum de les etapes necessàries per a la fabricació d'un circuit integrat

Un cop gravats tots els circuits en l'oblea, es comprova el procés de la fotolitografia. Si aquest és correcte, es parteix l'oblea de forma que cadascun dels circuits integrats quedi separat en daus individuals. Cadascun d'aquests daus és soldat sobre una càpsula on les sortides dels circuits es connecten mitjançant fils als electrodes que surten fora de la càpsula, i seguidament es fa un segellat al foc i el dispositiu ja és llest per a la seva comprovació final.

Recerca en VLSI

Quan hom sent parlar d'aquests avenços tècnics té tendència a creure que tot això passa lluny de nosaltres. Si bé és veritat que actualment encara hem d'importar la major part de la tecnologia, en particular de l'alta tecnologia, no ho és gens menys que aquí, i en particular a Catalunya, s'està intentant d'agafar aquest tren de la VLSI i no pas amb bitllet per a vagó de cua.

En el Departament d'Informàtica de la Universitat Autònoma de Barcelona existeix un grup de recerca en disseny VLSI integrat

per uns 15 professionals de la informàtica i la microelectrònica. Actualment, aquest grup interacciona amb d'altres de francesos i de belgues, com ara el centre IMAG de Grenoble, el LAAS de Toulouse i el centre de microelectrònica de la Universitat de Louvain, i conjuntament participen en un o dos MPC (Multiple Project Chip) per any. Cada un dels centres fa el disseny dels seus propis projectes que, un cop coordinats pel centre IMAG de Grenoble, són enviats als Estats Units de Nord-Amèrica per fer-ne les màscares corresponents.

El procés de fabricació es fa a Cenet o a Louvain o a Matra i tornen els chips corresponents a cada grup o institució. Un cop rebut el chip és testejat, utilitzant sofisticats aparells de mesura i simulació com ara un Tektronik 3270.

Quant als recursos tècnics, aquest equip de recerca compta amb dos ordinadors: un VAX 750 i un PDP 11/34 de DIGITAL, un plotter BENSON 1313 de tres plomes, dues pantalles gràfiques: una 4014 de Tektronik i una VT102, un digitalitzador de taula i un bon grapat de paquets de software d'ajut al disseny. És a dir, software de CAD: el LUCIE (editor gràfic), DIANA (simulador elèctric), LOGMOS (simulador lògic i temporal) entre d'altres.

Actualment, aquest grup ja porta dissenyats alguns circuits integrats tan en el món acadèmic com en el món industrial, i un dels objectius de futur és poder realitzar un MPC individual a finals d'aquest any o a principi de 1986.

Per altra banda, el Departament d'Electricitat i Electrònica de la mateixa Universitat disposa d'instal·lacions i d'equip humà per dur a terme la fabricació de circuits integrats. Per al procés de fabricació es necessiten uns equips i unes instal·lacions molt sofisticades i costoses. En particular es disposa de:

- una sala d'atmosfera controlada, dita **sala blanca**, que necessita estar a temperatura de 20 graus i a una humitat constant del 40 %, i que com a màxim pot tenir 5 partícules de pols per centímetre cúbic. En cas contrari es crearien impureses no desitjades sobre l'òxid de silici i el resultat seria inaprofitable. És per això que quan el director d'aquest equip m'ha ensenyat aquesta sala, m'ha fet cobrir les sabates amb unes plantofes fins als genolls i m'ha fet posar una bata verda i un barret blanc per al cabell. Tenia la impressió d'entrar a un quiròfan.
- un **depurador d'aigua**, perquè aquesta ha d'ésser ultra-pura en tot el procés de fabricació.
- els **elements químics** necessaris i **gasos** com ara oxigen, nitrògen, argó, etc., en el seu estat més pur.
- **forns de deposició** que operen sobre el silici a uns 1200 graus de temperatura fent passar gas amb les impureses que desitgem que s'hi quedin impregnades.
- **forns d'oxidació** que operen a uns 1000 graus de temperatura i permeten el pas d'oxigen per dur a terme l'obtenció de l'òxid de silici necessari en tot procés de fabricació de circuits integrats
- **equip de vaporització** de metall mitjançant una cambra de buit
- **equip d'encapsulat**, el qual permet fer les connexions de les sortides del dau

amb les potes del chip, mitjançant ones ultrasòniques i fils d'alumini o d'or. Aquest encapsulat es fa sobre resines d'apoxi o de ceràmica perquè permetin una bona dissipació d'energia.

Aquest és un dels processos de fabricació més sensibles i perfeccionats que ha pogut fer l'home. El seu preu a pagar és la puresa dels materials i les condicions extremes que hem ressenyat durant tot el procés de fabricació, a més del finançament dels equips. Per tenir una idea del seu cost penseu que només l'aparell d'obtenció de les màscares a partir del disseny val uns 200 milions de pessetes.

Si a més es vol disposar de l'aparell d'implantació iònica, que el que fa és treure electrons de l'última capa del silici i accelerar-los mitjançant uns corrents de 200.000 volts perquè en xocar-hi les impureses quedin impregnades, costa al voltant dels 150 milions de pessetes. Bé, amb tot aquest equip material de què es disposa (només manca l'equip d'implantació iònica) i amb l'esforç del grup de recerca i tècnica que treballa en aquest Departament (en total són vuit entre especialistes en microelectrònica i tècnics de fabricació) ja s'ha començat a fabricar algun circuit integrat, encara que en tecnologia PMOS. Però ben aviat pensen fer-ho amb tecnologia NMOS i CMOS. Aquests circuits tenem con a destinació tasques docents i d'investigació en la nostra pròpia Universitat i en d'altres. En un futur es pretén, a més, donar servei al món extrauniversitari. Crec que és important assenyalar que ambdós equips de recerca formen part del recentment creat Institut de Microelectrònica dins del marc d'acció del PEIN (Plan Electrónico e Informático Nacional). Ens sentim contents que aquí a Catalunya es participi en el desenvolupament d'aquestes tècniques, ja que ens poden portar més capacitat d'autoproducció microelectrònica i en conseqüència tenir menys dependències de les grans potències.

Informàtica a l'escola

Fragment del article publicat
a NOVATICA, n.º 61, Març 1985

Josep Casanoves, Pilar Conesa,
Diana Garrigosa, Xavier Kirchner

Els primers assaigs, en l'àrea d'utilització dels ordinadors dins del programa educatiu varen començar als anys seixanta augurant a l'ordinador el paper de «màquina que a la llarga substituirà al professor» en el procés de subministrar informació, plantejar preguntes, avaluar respostes i, en funció d'aquestes, donar nova informació o repetir, potser amb un altre nou enfocament, informacions anteriors.

Aquestes experiències s'han continuat fins al present en allò que s'ha denominat EAO (Ensenyament Assistit per Ordinador) àrea que pot ser dividida en dues branques que corresponen a diferents funcionalitats de l'ordinador.



(1) Ensenyament per transmissió d'informació alternada amb preguntes seguint les **tècniques d'ensenyament programat** originades per Skinner i altres.

(2) **Aprentatge per simulació** on l'ordinador, mitjançant un programa adequat, **reprodueix situacions reals que permeten a l'alumne «veure», de manera simulada, l'efecte de les seves accions**, bé sigui la simple representació d'una funció matemàtica de la qual fixa l'alumne els paràmetres, o la reproducció d'un vol espacial en què l'ordinador controla moviments de cabina, sorolls i imatges.

Ambdues tècniques són utilitzades i comercialitzades a bastament malgrat que, per raons del cost dels equips i del desenvolupament de programes, la seva àrea d'aplicació no experimental estigui limitada a la formació professional, en temes molt específics dins del marc de l'empresa (electrònica, banca), o en situacions en les quals es pretén la formació d'un gran nombre d'alumnes amb pocs professors o amb professors poc qualificatius, com el programa d'ensenyament de matemàtica, anglès i hebreu als nenes àrabs a Israel. Darrerament, la incorporació d'imatge «real» controlada per ordinador (video-disc o imatge generada digitalment) obre una via d'avenços prometedors en aquesta àrea.

Un món completament diferent apareix quan es planteja la utilitat pedagògica de l'ordinador «per se», es a dir, el valor pedagògic de l'activitat que anomenem, potser incorrectament, «informàtica».

A l'activitat informàtica poden atribuir-se-li quatre grans objectius pedagògics delimitables en quatre àrees amb diferent nivell d'importància,

- heurística* en el plantejament de situacions desconegudes,
- lògica* en l'anàlisi de situacions complexes,
- rigor* en la realització de les respostes i
- desmitificació* de la tecnologia (tant en el sentit de pèrdua de la por com de la fe cega).

El primer objectiu, «heurística en el plantejament de situacions desconegudes», enllaça amb el fet de la «velocitat» de la generació dels coneixements.

Malgrat que el coneixement no ha estat mai una realitat estàtica, al llarg de molts segles els avenços han estat prou lents per a permetre una absorció relativament lenta d'aquests per part del medi social. En el present, entre d'altres raons a causa de la rapidesa de les comunicacions i de la pressió del consum, els descobriments es produeixen, perfeccionen i utilitzen en forma generalitzada a intervals de temps cada cop més curts.

Davant d'aquest fet, s'imposa una revisió del sistema educatiu: **l'escola, tant o més que comunicar al nen un determinat món de coneixements, ha de plantejar-se de preparar-lo per a assimilar les noves i desconegudes realitats a les quals haurà de fer front al llarg de la seva vida, transmetent-li una positiva actitud de recerca**, cosa que vol dir perdre la por a enfrontar-se amb allò que es desconeix, que ningú no li ha ensenyat i saber aprendre dels propis i gairebé sempre inevitables errors. L'entorn escolar convencional és orientat cap a la penalització o com a mínim la desqualificació de les equivocacions.

L'alumne que comet errors treu pitjors notes o és pitjor considerat que el que no els comet. Això, que és més o menys just quan es tracta de reproduir «el que ha estat ensenyat», ho deixa de ser quan la situació en la qual s'ha produït l'error planteja una realitat desconeguda. En aquests casos treu més profit, pel que fa a l'aprenentatge, qui s'equivoca i sap analitzar i corregir els errors que qui encerta una mica per intuïció o atzar, i, evidentment, molt més que qui, amb la por d'equivocar-se, no actua.

La realització de projectes, que poden ser molt senzills, **amb ordinador és una activitat propensa a l'error**. Per una banda els programes no pertanyen al món de la realitat tangible i això representa un

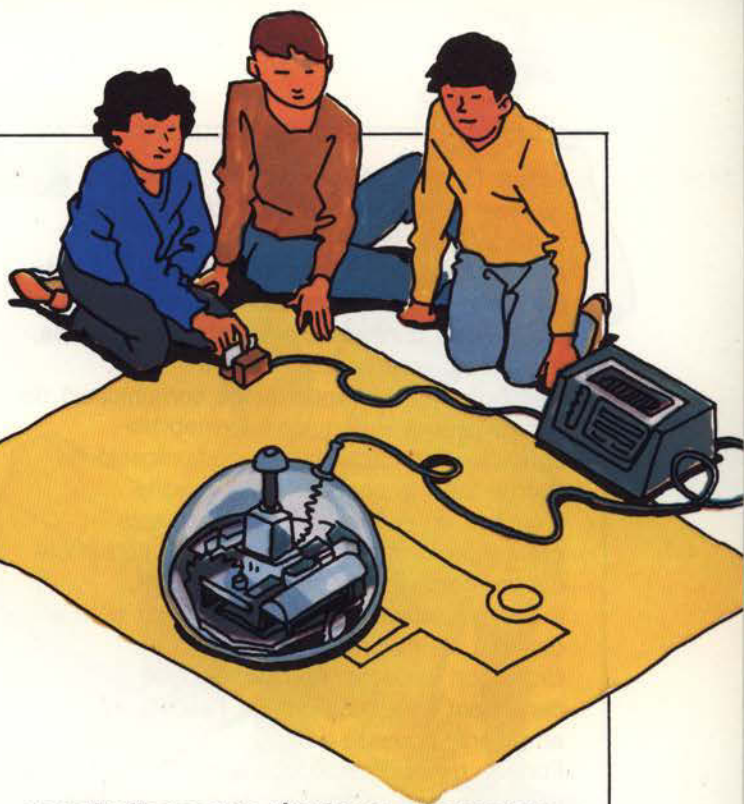
increment del nivell d'abstracció. Un sistema mecànic, per exemple, amb una certa capacitat espacial es pot imaginar en fase de disseny preveient les fallades abans que sigui construït. Imaginar el funcionament d'un programa, que mai no es veu funcionar «per dins», no és tan senzill. Una altra font d'errors és el fet que **l'ordinador no admet ambigüitats ni instruccions incompletes, i l'ésser humà no està acostumat a treballar en aquestes condicions. L'assaig intel·ligent i l'anàlisi i correcció de les equivocacions son realitats obligades.** A més a més l'ordinador, en comparació amb d'altres mitjans per a estimular l'actitud investigadora i creativa dels alumnes, té l'avantatge que permet el plantejament de problemes en molts diferents camps, no presenta especials perills ni es deteriora amb els assaigs i el seu cost de funcionament és molt baix.

El segon objectiu s'ha descrit com la «lògica en l'anàlisi de situacions complexes».

La rapidesa de les comunicacions, el teletractament i l'emmagatzemament massiu de les informacions donen lloc, cada cop més freqüentment, a problemes que ens desborden, no per la dificultat individual dels seus components sinó pel nombre d'aquests i de les seves interrelacions. En aquestes situacions no acostumen a funcionar les improvisacions ni les solucions parcials. El medi escolar ha de donar als alumnes, més que un conjunt de tècniques, difícils d'establir de manera generalitzada, una **actitud de superació de la complexitat mitjançant la fragmentació del problema segons una certa lògica.**

Per les raons apuntades abans, la resolució de problemes utilitzant ordinadors no és una tasca senzilla. Aquest fet ha portat els tècnics a plantejar-se les formes millors per a assolir problemes en la seva totalitat i dissenyar-ne algorismes de resolució vàlids.

Les tècniques utilitzades i sobretot, l'actitud mental induïda per l'exercici



repetit d'aquestes tècniques, representa un valor en sí mateix aplicable fora de l'àmbit informàtic o tecnològic.

El tercer objectiu, el «rigor en la realització de les respostes» no és un valor, com els anteriors, nascut a conseqüència del canvi tecnològic. Es podria dir que és un valor «tradicional». De totes maneres la seva importància s'ha vist augmentada en un món que tolera cada vegada menys l'ambigüitat i la improvisació.

Finalment el quart objectiu, la «desmitificació de l'ordinador», és un tema que ens interessa des de dos aspectes complementaris.

El primer és especialment important a l'hora de plantejar-se els mètodes de realització pràctica de l'experiència. És probable que només una petita part dels alumnes tinguin la informàtica com a professió; però també ho és que gairebé tots manipularan ordinadors en les seves activitats professionals o simplement socials, bé sigui per consultar una referència bibliogràfica, una història clínica, una corba de tendència econòmica o senzillament potser per fer

arribar la «llista de la compra» mitjançant el terminal de l'habitatge a l'«ordinador» del supermercat del barri.

En tant que els ordinadors incideixen o, més exactament, es responsabilitzen de la transmissió d'informacions entre els humans, rebent-les, emmagatzemant-les i distribuint-les, la capacitat de comunicació de l'home amb la «màquina-intermediari» condiona la capacitat de comunicació de l'home amb la societat que l'envolta.

L'actitud en què es produeix aquesta comunicació, tant en l'aspecte de l'absència de por com de fe cega és, per tant, important, i l'escola es pot plantejar com a objectiu de millorar-la.

Un segon aspecte, es que allò que certament s'ha de plantejar l'escola és no empitjorar aquesta actitud.

Formes de realització que associïn l'èxit en la utilització dels ordinadors al gust per àrees de coneixement determinades, poden portar una part dels alumnes a la conclusió que aquest assumpte «no és per a ells». Aquest perill no està tan lligat a l'associació dels ordinadors amb la matemàtica, àrea en la qual els alumnes tenen una formació similar, sinó a l'associació dels ordinadors amb la tecnologia, àrea generalment externa als programes escolars i tradicionalment reservada als alumnes de sexe masculí.

Fixats els objectius, cal definir la metodologia a seguir que no serà altra que la interacció alumne-ordinador en el cicle: assaig intel·ligent, anàlisi i aprenentatge de l'error.

El context general es podria descriure com a «resolució de problemes» o «realització de projectes», entenent com «problemes» o «projectes», en sentit ampli, situacions en què l'alumne pretén que l'ordinador faci alguna cosa concreta com pot ser un dibuix per la pantalla, un càlcul, la transformació d'una frase o l'anàlisi d'un text. El més important no és què es fa sinó com, i en aquest sentit com és especialment important l'assumpció per part de l'alumne del «repte»

que representa el projecte, fer-lo «seu», tant si ha estat ell mateix qui l'ha proposat, com si l'ha plantejat el professor o ha agafat la idea del treball d'altres alumnes. Tan important com l'assumpció personal del problema és la lliure i personal interacció alumne-ordinador.

En tant que els objectius que es plantegen fan referència a actituds, la realització de l'experiència ha de considerar-se a llarg termini, existint com a única restricció en la determinació de l'edat dels alumnes que comencen, que disposin dels **coneixements mínims per a poder plantejar-se projectes i la capacitat d'abstracció necessària per a treballar amb un ordinador.**

Els coneixements transmesos amb aquesta manera d'enfocar la introducció de la Informàtica a l'escola són bàsicament eines per analitzar i resoldre problemes en general, per a treballar amb mètode i rigor, i per a incitar a la investigació. Els plantejaments i objectius esmentats són vàlids en tots els moments del procés de formació de la persona i per tant és possible considerar un programa educatiu continuat des de la Bàsica al C.O.U. Caldrà trobar per a cada edat la metodologia i el llenguatge adient, però conceptualment només cal adequar l'èmfasi que es posa en un o altre objectiu en funció de l'evolució cognitiva de l'alumne. En efecte, analitzant el procés escolar s'observa que hi ha un canvi bruscat en la capacitat mental dels alumnes aproximadament als **13-14 anys**, fet que ve reflectit en els programes educatius pel canvi d'etapa EGB-BUP. A aquesta edat apareix la **capacitat d'abstracció lògica**, indispensable per a desenvolupar l'**actitud analítica** que s'ha descrit dins dels objectius del segon grup. Un canvi de metodologia és desitjable i fa possible estructurar el programa en dos cicles: un primer en el que cal donar especial importància als objectius de tipus heurístic i un segon cicle, que correspondria als cursos de B.U.P.-C.O.U. on s'accentuen més els aspectes lògics i analítics.

Recull de basic



Quan hom es troba amb la necessitat de fer un programa pot passar que conegui quin tipus de sentència cal fer servir, per exemple: ens cal una sentència de repetició o bé de bifurcació..., però no sapiga el nom concret de la sentència que s'ha d'usar. D'altres vegades el que passa és que hom coneix la instrucció que defineix la sentència però no recorda o no sap quin és el seu format o la seva sintaxi.

Per a resoldre situacions del primer tipus només cal fullejar les parts de BASIC dels llibrets que ja teniu. Amb l'ànim d'ajudar al programador que, en un moment determinat, necessita una informació ràpida de qualsevol de les paraules BASIC que ja us hem comentat hem decidit d'agrupar-les en ordre alfabètic i col·locar-les en una taula

especificant-ne format, funció i el número de pàgina i llibret on podeu trobar-ne més informació. A fi d'ésser el més complets possible hem inclòs una columna d'observacions en la que hi possem notes i comentaris.

45

PARAULA DEL BASIC	FUNCIÓ	FORMAT	OBSERVACIONS	SITUACIÓ
-------------------	--------	--------	--------------	----------

ABS	Elimina el signe del número.	ABS (número)	Transforma en positius els números negatius.	IV-49
AND	Es un operador lògic que vol dir "i".	cond1 AND cond2	Apareix dins els IF...THEN	1-77
ASC	Dóna el codi ASCII (decimal) del primer caràcter d'una cadena.	ASC (cadena)	En els ZX de Sinclair es diu: CODE	III-31
AT	Posicionament en la pantalla. (ZX-Spectrum).	AT núm. fila, núm. columna	En el PC d'IBM és LOCATE, i en altres s'usa l'a per posicionar-se: afile+columna.	II-34

BEEP	Genera sons.	BEEP durada, to	To selecciona la nota i durada dóna el temps en segons en que aquesta està activada.	II-55
BIN	Serveix per treballar amb valors numèrics, però expressats en binari.	BIN núm. binari	útil per definir nous caràcters per dibuixar.	II-34
BORDER	Dóna un color determinat al contorn del rectangle de dibuix.	BORDER núm. color	núm. color pot ésser una expressió numèrica qualsevol.	II-31
BREAK	Talla l'execució d'un programa des del teclat.	BREAK	És una tecla.	II-39
BRIGHT	Serveix per destacar el text, fent que brilli.	BRIGHT 1 o 0	Amb 1 activem la brillantor, amb 0 la desactivem.	II-32
CIRCLE	Dibuixa circumferències.	CIRCLE columna, línia, radi	Els valors de columna i línia donen el centre de la circumferència.	II-39
CLOAD	És el mateix que LOAD, però per al micro Dragon.	CLOAD «nom del programa».		I-86
CLOSE	Tanca el fitxer que està obert.	CLOSE N	N es el número de fitxer.	IV-45
CLS	Esborra la pantalla.	CLS	En el micro Dragon es tria el color de la pantalla tot posant el seu número darrera de CLS.	I-73
CODE	Dóna el codi ASCII (decimal) del primer caràcter d'una cadena.	CODE (cadena)	Hi ha micros en que es diu: ASC.	III-31
COS	Calcula el cosinus d'un angle expressat en radians.	COS (angle).	La seva inversa és la funció arcocinus: ACS quan $-PI < x <= PI$	IV-50
CSAVE	Fa el mateix que SAVE, però per al micro Dragon.	CSAVE «nom del programa».		I-86
DATA	Diu a l'ordinador que tot el que hi ha darrera seu fins a la següent instrucció són dades.	DATA dada1,..., dada N	Va sempre acompanyada de la READ.	III-49
DELETE	Permet esborrar caràcters.	DEL	Es una tecla.	I-44

DIM	Serveix per definir el tamany i la quantitat d'índexs d'una variable indexada.	DIM nom variable (ind1,...,indN)	En el ZX-Spectrum, quan s'ha de treballar amb variables indexades alfanumèriques, és necessari afegir un índex al final, per dir la longitud màxima de les cadenes.	II-13
DRAW	Dibuixa una línia d'una tirada des de l'últim punt dibuixat.	DRAW salt línia, salt columna, gir	Gir es optatiu; si no es possa, la línia serà recta.	II-38
END	Finalitza l'execució d'un programa.	END	No tots els micros la tenen. (Usen només l'STOP).	I-70
ENTER	Indica a l'ordinador que ja pot llegir les dades que hem escrit, tant si són instruccions d'un programa, com dades d'entrada.	ENTER	És una tecla.	I-44
EOF	Significa END OF FILE, és a dir, final de fitxer.	EOF (N).	N és el número de fitxer.	IV-45
FIELD...AS.	Assigna a un fitxer el seu buffer.	FIELD num, ... AS ...	Permet definir-nos el registre.	IV-47
FIX	Torna el valor sencer resultant de treure la part decimal.	FIX (número).	Per números positius coincideix amb INT.	IV-49
FLASH	Fa que el que s'ha d'escriure o dibuixar surti fent intermitències.	FLASH 1 o 0.	Si posem 1 activarem les intermitències; amb 0 les desactivarem.	II-32
FOR...TO	Estructura de repetició. Tot el que hi haigui entre el FOR i la instrucció NEXT es repetirà fins que el comptador prengui el valor final.	FOR comptador = valor inicial. TO valor final.	Comptador ha d'ésser una variable, mentre que valor final i inicial poden ésser expressions numèriques qualsevols. Comptador es pot modificar dins un bucle.	II-11
GET	Comprova si s'està pitjant alguna tecla. En cas afirmatiu guarda el seu significat a la variable.	GET variable\$.	És una funció útil per jocs. En altres micros es diu INKEY\$.	III-26

GET	S'usa per llegir un registre d'un fitxer.	GET N, NR	Una vegada llegit el registre, transfereix les dades al buffer. N és núm. de fitxer. NR núm. de registre.	IV-47
GOSUB	Per executar una subrutina, després el programa torna a la instrucció següent a la del GOSUB.	GOSUB núm. línia.	Núm. línia és on hi ha la primera instrucció de la subrutina.	III-23
GOTO	Bifurcació incondicional a una altra part d'un programa.	GOTO núm. línia.		I-76
IF...THEN	Defineix la sentència de condició. Vol dir: Si succeeix tal cosa aleshores fer...	IF condició THEN llista d'accions.	També hi ha l'opció de posar un grup d'accions alternatiu en el cas de que la condició no es compleixi. N'hi hauria prou afegint: ...ELSE llista d'accions.	I-76
INK	Defineix el color de la «tinta» amb que s'escriurà o dibuixarà.	INK núm. color.	núm. color pot ésser una expressió numèrica qualsevol.	II-31
INKEY\$	Comprova si s'està pitjant alguna tecla. En cas afirmatiu guarda el seu significat.	INKEY\$	És una funció útil per jocs. En el Commodore es diu GET.	III-26
INPUT	Espera dades entrades per teclat i les emmagatzema en les variables especificades.	INPUT llista de variables.	Pot emetre un missatge per pantalla posant-lo entre (") després d'INPUT.	I-70
INT	Dóna la part sencera d'un decimal.	INT (núm. dec.)		IV-48
KILL	Esborra un fitxer d'un disc.	KILL "nom del fitxer".	Únicament es pot utilitzar amb unitat de disc o floppy.	IV-46
LEFT\$	Ens dóna els n primers caràcters d'una cadena.	LEFT\$ (variable\$,n)	En el ZX-Spectrum es posa: variable\$ (TO n)	III-29
LEN	Dóna la longitud d'una cadena.	LEN (variable\$)		III-29
LET	Carregar en una variable o bé un valor concret, o bé el contingut d'una altra variable, o bé el resultat d'una expressió.	LET variable = valor a assignar.	En molts micros n'hi ha prou amb posar l'assignació sense el LET.	I-75

LIST	Escriure a pantalla tot, o part del programa que tinguem a la RAM.	LIST lín 1-lín 2	Es poden obviar o un, o els dos paràmetres.	I-74
LOAD	Carrega un programa a la RAM.	LOAD "nom del programa".	Cal pitjar la tecla "Play" després.	I-86
LOC	Dóna l'últim registre amb el que hem treballat.	LOC (N), N és el número de fitxer.	Per fitxers d'accés directe.	IV-46
LSET	Per assignar els valors als buffers de sortida.	LSET variable\$=variable\$	Assigna a la variable els valors per l'esquerra.	IV-47
MID\$	Extreu caràcters d'una cadena.	MID\$ (variable\$, X,Y)	Y serà el nombre de caràcters que desitgem obtenir a partir de la posició X de la cadena. Al ZX-Spectrum s'escriu: variable\$(X TO Y).	III-28
NAME..AS.	Canvia el nom d'un fitxer.	NAME nom antic AS nom nou.		IV-46
NEW	Esborra tota la memòria RAM.	NEW	Cal posar-la abans d'escriure un nou programa.	I-70
NEXT	Incrementa, o decrementa, segons el paràmetre de STEP, el comptador d'un bucle FOR.	NEXT comptador.		II-11
NOT	És un operador lògic que vol dir "no".	NOT condició.	Apareix dins els IF...THEN	I-77
ON	Possibilita triar la part de programa a executar mitjançant "menús".	ON expressió GOTO núm. línia ON expressió GOSUB núm. línia	Segons el valor d'expressió: 1,2,...,n el programa anirà al primer, segon,... núm. línia.	III-25
OPEN	Obre un fitxer.	OPEN "tipus de fitxer", núm. de canal, "nom del fitxer", longitud de registre.		IV-45
OR	És un operador lògic que vol dir "o".	cond1 OR cond2	Apareix dins els IF...THEN	I-77
PAPER	Tria el color de fons de la pantalla pel que s'escriu a partir d'ella.	PAPER núm. color.	Seguida de CLS, deixa tota la pantalla del mateix color. Pel Commodore POKE 53281, núm. color.	II-31

PAUSE	Provoca una espera durant l'execució.	PAUSE expressió.	El valor d'expressió determina la durada de l'espera. Un segon = 50.	II-35
PEEK	Mostra què hi ha en una adreça de memòria determinada.	PEEK adreça		II-33
PLOT	Dibuixa un punt a la pantalla.	PLOT núm. línia, núm. columna	El punt (0,0) es a baix a l'esquerra. Alta resolució.	II-37
POKE	Carrega un valor numèric en una adreça de memòria.	POKE adreça.	Útil per a definir nous caràcters.	II-33
PRINT	Escriure quelcom per pantalla.	PRINT llista (on llista tant poden ésser variables com constants).	Sense paràmetres fa un salt de línia.	I-71
PUT	Grava el registre que hi ha en el buffer.	PUT unitat, núm. de registre.		IV-47
READ	Carrega a les variables les dades de dins un DATA.	READ variables.	Acompanya a un DATA.	III-48
REM	Per introduir comentaris aclaridors dins un programa.	REM comentari.	És convenient usar-lo, sobre tot a l'inici de les subrutines.	I-69
RSET	Assigna valors als buffers de sortida.	RSET variable\$ = variable\$	Assigna els valors a la variable per la dreta.	
RESTORE	Gestiona les lectures que es fan sobre DATA.	RESTORE núm. línia	Cal posar-la quan hi ha més d'un DATA.	III-49
RETURN	Marca la fi de la subrutina. Fa que el programa vagi a executar la instrucció següent al GOSUB que ha cridat la subrutina on es troba aquest RETURN.	RETURN		III-23
RIGHT\$	Equivalent a la LEFT\$ però començant per la dreta.	RIGHT\$ (variable, n)	El ZX-Spectrum no té cap funció similar.	III-29
RND	Funció per generar números aleatoris entre a i b.	a+RND*(b-a).	Per a Dragon, la notació es: a+RND(b-a)	I-83
RUN	Executar un programa	RUN	Pot dur com a paràmetre el número de línia on s'ha de començar l'execució.	I-70

SAVE	Salvament de programes. Serveix per a desar un programa que estigui a la RAM.	SAVE "nom del programa"	S'han de pitjar les tecles "Play" i "Record" de l'enregistrador abans d'executar-la.	I-86
SCREEN\$	Obté el caràcter que ocupa una posició determinada de la pantalla.	SCREEN\$ (X,Y)	X és el número de línia i Y el de columna.	III-27
SIN	Calcula el sinus d'un angle expressat en radians.	SIN (angle)	La seva inversa es la funció arcsinus: ASN quan $-\pi \leq X \leq \pi$	IV-50
STEP	Expressa l'increment, o decrement, que fa variar el comptador d'una instrucció FOR.	STEP expressió.		II-12
STOP	Serveix per a interrompre l'execució d'un programa amb la possibilitat de tornar-lo a engegar.	STOP		III-23
STR\$	Converteix un valor numèric en la cadena numèrica corresponent.	STR\$ (valor numèric).	VAL és la inversa de STR\$ però no al revés.	III-30
STRING\$	Permet posar dins una variable alfanumèrica una cadena repetida N vegades.	STRING\$ (N,C)	C és la cadena que volem escriure N vegades. Pot ésser en codi ASCII o bé entre ("). El ZX-Spectrum i Commodore no tenen res semblant.	III-29
TAN	Calcula la tangent d'un angle expressat en radians.	TAN (angle)	La seva inversa és la funció arc tangent: ATN quan $(-\pi/2) \leq X \leq (\pi/2)$	IV-50
VAL	Converteix una cadena numèrica en la quantitat numèrica que representa.	VAL (cadena numèrica).		III-30
WRITE	Permet escriure en un fitxer seqüencial	WRITE #N, var1,... ...,varM	N és l'identificador numèric del fitxer i var J les variables on hi ha el que s'ha d'escriure.	IV-45
	Serveixen per separar sentències dins una mateixa línia de programa.			I-70

Altres

programmes

El joc del parxís

S'ha escrit en forma de programa el tant conegut joc del parxís. És un programa llarg però interessant doncs recull gairebé totes les instruccions de BASIC que s'han explicat. Un cop analitzat el programa, es pot pensar que hi ha altres formes diferents de fer-lo. És bo que això passi, doncs voldrà dir que ja s'ha adquirit un estil propi de programar. Sempre és un bon exercici l'intentar escurçar un programa tot assolint els mateixos resultats.

Si es disposa d'un micro diferent del que s'ha usat per confeccionar el programa, és interessant de crear la pròpia versió doncs això dóna agilitat i flexibilitat a l'hora de programar.

Comentarem seguidament, pas a pas, el que es va fent:

La línia 10 posa la pantalla en blanc. A les línies 20, 30 i 40 es defineixen les taules N, S i K. La taula N (99,3) emmagatzema els números de les caselles i les coordenades que aquests ocupen a la pantalla. La taula S (104,11) conté la informació que seguidament detallarem sobre cadascuna de les 104 caselles del parxís que es fan servir al llarg del joc. Els 11 elements que tenim per a cada casella són:



1 i 2: coordenades de la part esquerra, o de la superior segons l'orientació de la casella.

3: caràcter que configura aquesta part de la casella.

4 i 5: inicialment contenen un zero, però al llarg del joc s'usaran per contenir el número de fitxa i el número del jugador.

6 i 7: coordenades de la part dreta, o de la inferior segons l'orientació, de la casella.

8: caràcter que configura aquesta part de la casella.

9 i 10: com 4 i 5.

11: conté el número corresponent al color de la casella.

La taula K (4,4,5) s'usa per a controlar les fitxes. El primer índex es correspon amb el número del jugador, el segon amb el número de la fitxa, i el darrer conté la següent informació:

1 i 2: indiquen les coordenades de la fitxa quan és a «casa».

3: conté el número de la fitxa.

4: conté el color.

5: conté el número de la casella on està situada. Aquest camp va variant al llarg del joc.

Per exemple, si en un moment donat del programa volem saber on hi ha la fitxa 3 del jugador 2, haurem de posar K (2,3,5).

La configuració de les caselles queda definida des de la línia 100 a la 470 de la següent manera:

100-170: caràcter de la part esquerra.

200-270: caràcter de la part dreta.

300-370: caràcter superior.

400-470: caràcter inferior.

Entre les línies 490 y 690 hi ha uns bucles que permeten fer el dibuix del taulell. Hi ha uns quants IF's que controlen les coordenades on cal canviar de color, per exemple, a la línia 516, si les coordenades de la casella són 4 i 12, la variable Z canvia del color 7 al 6. Una vegada dibuixat, la Z recupera el valor inicial, en el nostre cas 7. A les línies 900-930 el que fem és carregar la taula N (99,3) amb els números i les posicions que aquests ocupen a la pantalla,

per això usem els DATA que hi ha entre les línies 5000 i 5310. Seguidament, fins la 960, escribim els números al taulell. De la línia 970 a la 1130 acabem de pintar el parxís, en concret les «cases» i la part central:

La «casa» del jugador 1 de color groc (970-978).

La «casa» del jugador 2 de color blau (980-988).

La «casa» del jugador 3 de color verd (990-998).

La «casa» del jugador 4 de color vermell (1000-1008).

Les dades que hi ha emmagatzemades en els DATA que es troben entre les línies 6000 i 6288 són carregades a la taula S, per fer-ho usem els dos bucles que hi ha entre les línies 1200 i 1260. El mateix fem amb la taula K a les línies 1300-1350.

Els números de les fitxes que inicialment apareixen a les «cases» són impresos amb les línies 1400-1430.

Fins aquí el que hem fet ha estat únicament el dibuix inicial del taulell. Ara, amb tot a punt, es pot començar a jugar.

Per controlar el torn dels jugadors usem les línies 2000-2190. A la 2170 generem aleatòriament un número que es correspon amb una tirada de dau. Per saber qui és qui ha tirar el dau, el seu número fa pampallugues.

A la línia 2192 es controla si ha sortit un 5, en cas afirmatiu el programa passa a la subrutina de la línia 2225; en cas contrari es mira si ha sortit un 6, en cas afirmatiu el programa passa a la subrutina de la línia 4000, en cas contrari salta a la línia 2600. De la línia 2225 a la 2600 es controla tot el referent al 5: si encara queden fitxes per sortir de «casa», si a la casella de sortida hi ha dues fitxes del mateix jugador, o bé de jugadors diferents, doncs aleshores cal «matar» a l'últim que hi ha arribat. A partir de la línia 2600 es controla el número de fitxa que es vol moure, es comproba que la fitxa no tingui cap obstacle, mirant si hi ha barreres o si cal matar en el cas que la

casella on acaba la jugada no sigui d'assegurança. Quan s'ha de matar, el programa salta a la subrutina de la línia 3500. En cas contrari es passa el torn al jugador següent.

El control del recorregut de les fitxes es fa de la següent manera:

Jugador 1.- de la 2700 a la 2795

Jugador 2.- de la 2800 a la 3016

Jugador 3.- de la 3020 a la 3156

Jugador 4.- de la 3160 a la 3296

A la línia 3400 s'efectua el moviment de la fitxa. La subrutina que hi ha a la línia 3650 serveix per contar els 20 punts d'haver matat.

Tot el referent al 6 es controla a la subrutina de la línia 4000: tres sisos seguits, si s'ha d'obrir barrera, etc.. A la 4500 es controla la posició de les fitxes quan són a la casella d'arribada.

A la línia 4660 hi ha una subrutina que ens demana amb quina fitxa volem contar els deu punts obtinguts en entrar fitxa.

Versió Spectrum

```
5 BORDER 1
10 CLS
20 DIM N(99,3)
30 DIM S(104,11)
40 DIM K(4,4,5)
50 LET d=0: LET h=0: LET u=0
51 PRINT AT 10,9: FLASH 1: INK 2:"JOC DEL PARXIS"
52 PAUSE 200: CLS : PRINT AT 2,8:"EL TORN PER MOURE FITXA VE DONAT PEL NUMERO
QUE ESTIGUI FENT PAMPALLUGUES."
53 PRINT AT 6,8:"PER SABER EL NUMERO QUE HA DONAT EL 'DAU' HEU D'APRETAR EL VO
STRE NUMERO DE JUGADOR"
54 PRINT AT 10,8:"QUAN VOLGUEU SALTAR EL TORN D'ALGUN JUGADOR POSEU 'Ø' EN
EL LLOC ON DEMANI FITXA PER JUGAR"
55 PRINT AT 21,9: INK 3:"APRETEU <ENTER>"
56 INPUT V#
57 CLS
100 POKE USR "b", BIN 11111111
110 POKE USR "b"+1,BIN 10000000
120 POKE USR "b"+2,BIN 10000000
130 POKE USR "b"+3,BIN 10000000
140 POKE USR "b"+4,BIN 10000000
150 POKE USR "b"+5,BIN 10000000
160 POKE USR "b"+6,BIN 11111111
170 POKE USR "b"+7,BIN 00000000
200 POKE USR "c", BIN 11111111
210 POKE USR "c"+1,BIN 00000001
220 POKE USR "c"+2,BIN 00000001
230 POKE USR "c"+3,BIN 00000001
240 POKE USR "c"+4,BIN 00000001
250 POKE USR "c"+5,BIN 00000001
260 POKE USR "c"+6,BIN 11111111
270 POKE USR "c"+7,BIN 00000000
300 POKE USR "d", BIN 11111110
310 POKE USR "d"+1,BIN 10000010
320 POKE USR "d"+2,BIN 10000010
330 POKE USR "d"+3,BIN 10000010
340 POKE USR "d"+4,BIN 10000010
350 POKE USR "d"+5,BIN 10000010
360 POKE USR "d"+6,BIN 10000010
370 POKE USR "d"+7,BIN 10000010
400 POKE USR "e", BIN 10000010
410 POKE USR "e"+1,BIN 10000010
420 POKE USR "e"+2,BIN 10000010
430 POKE USR "e"+3,BIN 10000010
440 POKE USR "e"+4,BIN 10000010
450 POKE USR "e"+5,BIN 10000010
460 POKE USR "e"+6,BIN 10000010
```



```

470 POKE USR "e"+7,BIN 1111110
490 LET Z=7
500 FOR q=12 TO 16 STEP 2
510 FOR i=0 TO 7
515 IF q=14 THEN LET Z=6
516 IF i=4 AND q=12 THEN LET Z=6
518 IF i=0 AND q=14 THEN LET Z=3
519 IF i=4 AND q=16 THEN LET Z=3
520 PRINT AT i,q; PAPER Z;CHR# (145);CHR# (146)
525 LET Z=7
530 NEXT i: NEXT q
540 LET Z=7
550 FOR q=2 TO 9
560 FOR i=8 TO 12 STEP 2
565 IF i=10 THEN LET Z=5
566 IF i=12 AND q=6 THEN LET Z=5
568 IF i=10 AND q=2 THEN LET Z=3
569 IF i=8 AND q=6 THEN LET Z=3
570 PRINT AT i,q; PAPER Z;CHR# (147);
580 PRINT AT i+1,q; PAPER Z;CHR# (148);
585 LET Z=7
590 NEXT i: NEXT q
600 FOR q=12 TO 16 STEP 2
610 FOR i=14 TO 21
615 IF q=14 THEN LET Z=2
616 IF q=16 AND i=17 THEN LET Z=2
618 IF q=14 AND i=21 THEN LET Z=3
619 IF q=12 AND i=17 THEN LET Z=3
620 PRINT AT i,q; PAPER Z;CHR# (145);CHR# (146)
625 LET Z=7
630 NEXT i: NEXT q
650 FOR q=20 TO 27
660 FOR i=8 TO 12 STEP 2
665 IF i=10 THEN LET Z=4
666 IF i=8 AND q=23 THEN LET Z=4
668 IF i=10 AND q=27 THEN LET Z=3
669 IF i=12 AND q=23 THEN LET Z=3
670 PRINT AT i,q; PAPER Z;CHR# (147);
680 PRINT AT i+1,q; PAPER Z;CHR# (148);
685 LET Z=7
690 NEXT i: NEXT q
900 FOR I=1 TO 99
910 FOR A=1 TO 3
920 READ X
925 LET N(I,A)=X
930 NEXT A: NEXT I
940 FOR I=1 TO 99
950 PRINT AT N(I,1),N(I,2);N(I,3)
960 NEXT I
970 FOR O=1 TO 8
972 FOR P=0 TO 5
974 PRINT AT P,O; PAPER 6;CHR# (128)
976 NEXT P
978 NEXT O
980 FOR O=1 TO 8
982 FOR P=16 TO 21
984 PRINT AT P,O; PAPER 5;CHR# (128)
986 NEXT P
988 NEXT O
990 FOR O=21 TO 28
992 FOR P=0 TO 5
994 PRINT AT P,O; PAPER 4;CHR# (128)
996 NEXT P
998 NEXT O
1000 FOR O=21 TO 28
1002 FOR P=16 TO 21
1004 PRINT AT P,O; PAPER 2;CHR# (128)

```

```

1006 NEXT P
1008 NEXT O
1010 FOR O=30 TO 31
1012 FOR P=0 TO 21
1014 PRINT AT P,0; PAPER 1;CHR# (128)
1016 NEXT P
1018 NEXT O
1020 FOR O=8 TO 9
1022 PRINT AT O,14; PAPER 6;CHR# (128);CHR# (128);
1024 NEXT O
1025 FOR O=10 TO 11
1027 PRINT AT 10,0; PAPER 5;CHR# (128)
1028 PRINT AT 11,0; PAPER 5;CHR# (128)
1029 NEXT O
1030 FOR O=12 TO 13
1032 PRINT AT O,14; PAPER 2;CHR# (128);CHR# (128)
1034 NEXT O
1040 FOR O=18 TO 19
1042 PRINT AT 10,0; PAPER 4;CHR# (128)
1043 PRINT AT 11,0; PAPER 4;CHR# (128)
1045 NEXT O
1100 PRINT AT 1,1; PAPER 6;"N.1"
1110 PRINT AT 20,1; PAPER 5;"N.2"
1120 PRINT AT 1,25; PAPER 4;"N.4"
1130 PRINT AT 20,25; PAPER 2;"N.3"
1200 REM Carrega taula caselles
1210 FOR I=1 TO 104
1220 FOR J=1 TO 11
1230 READ x
1240 LET S(I,J)=x
1250 NEXT J
1260 NEXT I
1300 FOR o=1 TO 4
1310 FOR p=1 TO 4
1320 FOR q=1 TO 5
1330 READ x
1340 LET k(o,p,q)=x
1350 NEXT q: NEXT p: NEXT o
1400 FOR o=1 TO 4
1410 FOR p=1 TO 4
1420 PRINT AT k(o,p,1),k(o,p,2); PAPER k(o,p,4);p
1430 NEXT p: NEXT o
2000 REM Control del numero de jugador
2005 IF d=1 OR d=2 THEN GO TO 2110
2010 PRINT AT 1,1; PAPER 6; FLASH 0;"N.1"
2020 PRINT AT 20,1; PAPER 5; FLASH 0;"N.2"
2030 PRINT AT 20,25; PAPER 2; FLASH 0;"N.3"
2040 PRINT AT 1,25; PAPER 4; FLASH 0;"N.4"
2100 LET h=h+1
2102 IF h>4 THEN LET h=h-4
2110 IF H=1 THEN PRINT AT 1,1; PAPER 6; FLASH 1;"N.1"
2120 IF H=2 THEN PRINT AT 20,1; PAPER 5; FLASH 1;"N.2"
2130 IF H=3 THEN PRINT AT 20,25; PAPER 2; FLASH 1;"N.3"
2140 IF H=4 THEN PRINT AT 1,25; PAPER 4; FLASH 1;"N.4"
2150 IF INKEY#="" THEN GO TO 2130
2160 IF INKEY#(<>STR$(H)) THEN GO TO 2130
2170 LET L=1+INT (RND*6)
2175 IF L<>6 THEN LET d=0
2180 PRINT AT 10,14;L
2185 FOR f=1 TO 4
2187 IF k(h,f,5)<>0 THEN GO TO 2210
2190 NEXT f
2191 REM Comprovar si es un 5
2192 IF l=5 THEN GO TO 2225
2194 REM Control de la possibilitat de tres 6 sense fitxes en joc
2195 IF l<>6 THEN LET d=0: GO TO 2000
2196 IF d=3 THEN LET d=0: GO TO 2000

```



```

2197 LET d=d+1: GO TO 2000
2200 REM
2205 LET i=0
2210 IF 1<>5 AND 1<>6 THEN GO TO 2600
2220 IF 1=6 THEN GO TO 4000
2225 REM Calcul de 5
2227 LET m=0
2230 FOR i=1 TO 4
2240 IF k(h,i,5)=0 THEN LET m=1: GO TO 2270
2250 NEXT i
2260 GO TO 2600
2270 LET i=5+(17*(h-1))
2280 IF S(i,4)<>0 AND S(i,9)<>0 THEN GO TO 2510
2290 IF S(i,4)=0 OR S(i,9)=0 THEN GO TO 2490
2300 REM Colocar una fitxa en la sortida
2310 INPUT "Quina fitxa vols moure";w
2320 IF w<0 OR w>4 THEN GO TO 2310
2330 IF w=0 THEN GO TO 2000
2335 IF k(h,w,5)<>0 THEN GO TO 2310
2340 PRINT AT k(h,w,1),k(h,w,2); PAPER k(h,w,4);CHR# (128)
2350 LET k(h,w,5)=i
2360 PRINT AT s(i,1),s(i,2); PAPER k(h,w,4);w
2470 LET s(i,4)=w: LET s(i,5)=h
2480 GO TO 2000
2490 REM Coloquem fitxa en sortida al costat d'una altra
2492 INPUT "Quina fitxa vols moure";w
2493 IF w<0 OR w>4 THEN GO TO 2492
2495 IF w=0 THEN GO TO 2000
2498 IF k(h,w,5)<>0 THEN GO TO 2490
2500 IF S(i,4)=0 THEN GO TO 2340
2502 PRINT AT k(h,w,1),k(h,w,2); PAPER k(h,w,4);CHR# (128)
2504 LET k(h,w,5)=i
2506 PRINT AT s(i,6),s(i,7); PAPER k(h,w,4);w
2508 LET s(i,9)=w: LET s(i,10)=h: GO TO 2000
2510 REM Seleccio de la jugada quan hi ha dues fitxes en la mateixa casella
2512 IF s(i,5)=h AND s(i,10)=h THEN GO TO 2530
2514 IF s(i,10)<>h THEN GO TO 3510
2516 GO TO 3540
2530 INPUT "Quina fitxa vols moure ";w
2540 IF w<0 OR w>4 THEN GO TO 2530
2550 IF w=0 THEN GO TO 2000
2560 IF k(h,w,5)=0 THEN GO TO 2530
2570 GO TO 2620
2600 INPUT "Quina fitxa vols moure";w
2610 IF w<0 OR w>4 THEN GO TO 2600
2615 IF w=0 THEN GO TO 2000
2617 IF k(h,w,5)=0 AND 1<>5 THEN GO TO 2600
2620 LET u=0
2640 IF h=1 THEN LET v=k(h,w,5): LET z=v+1: GO TO 2700
2650 IF h=2 THEN LET v=k(h,w,5): LET z=v+1: GO TO 2800
2660 IF h=3 THEN LET v=k(h,w,5): LET z=v+1: GO TO 3020
2670 IF h=4 THEN LET v=k(h,w,5): LET z=v+1: GO TO 3160
2680 STOP
2700 REM Control de barreres en la jugada del N.1
2705 IF v=0 THEN LET v=5
2710 IF z<=76 AND v<76 THEN GO TO 2750
2730 GO TO 3300
2750 FOR e=v+1 TO v+1
2755 LET i=e
2760 IF s(e,4)=0 THEN GO TO 2780
2770 IF s(e,5)=s(i,10) THEN GO TO 3300
2780 NEXT e
2790 IF i=76 THEN GO TO 4500
2795 GO TO 3400
2800 REM Control de barreres en la jugada de N.2
2805 IF v=0 THEN LET v=22
2810 IF z<=88 AND v<88 THEN GO TO 2835

```

```

2830 GO TO 3300
2835 IF v>68 OR v<18 THEN LET u=1
2840 FOR e=v+1 TO v+1
2850 LET i=e
2860 IF i>22 AND e<68 AND u=0 THEN GO TO 2890
2865 IF i>68 AND u=1 THEN GO TO 2890
2870 IF i>68 AND u=0 THEN LET i=i-68
2875 IF i=68 THEN GO TO 2890
2880 IF i>17 THEN LET i=i+60
2890 IF s(i,4)=0 THEN GO TO 2905
2900 IF s(i,5)=s(i,10) THEN GO TO 3300
2905 IF i>85 THEN GO TO 3300
3010 NEXT e
3015 IF i=85 THEN GO TO 4500
3016 GO TO 3400
3020 REM Control de barreres en la jugada de N.3
3025 IF v=0 THEN LET v=39
3030 IF z<=94 AND v<94 THEN GO TO 3060
3050 GO TO 3300
3060 IF v<35 THEN LET u=1
3070 FOR e=v+1 TO v+1
3080 LET i=e
3090 IF i>39 AND i<68 AND u=0 THEN GO TO 3130
3095 IF i>86 AND i<95 THEN GO TO 3130
3100 IF i>68 THEN LET i=i-68: GO TO 3130
3105 IF i=68 THEN GO TO 3130
3110 IF i<35 THEN GO TO 3130
3120 IF i>34 AND u=1 THEN LET i=i+52
3130 IF s(i,4)=0 THEN GO TO 3145
3140 IF s(i,5)=s(i,10) THEN GO TO 3300
3145 IF i>94 THEN GO TO 3330
3150 NEXT e
3155 IF i=94 THEN GO TO 4500
3156 GO TO 3400
3160 REM Control de barreres del jugador N.4
3165 IF v=0 THEN LET v=56
3170 IF z<=103 AND v<103 THEN GO TO 3200
3200 IF v>31 AND v<52 THEN LET u=1
3210 FOR e=v+1 TO v+1
3220 LET i=e
3230 IF i>56 AND i<68 AND u=0 THEN GO TO 3270
3245 IF i=51 THEN GO TO 3270
3260 IF i>51 AND u=1 THEN LET i=i+44: GO TO 3270
3264 IF i>68 AND i<96 THEN LET i=i-68
3270 IF i>103 THEN GO TO 3300
3275 IF s(i,4)=0 THEN GO TO 3290
3280 IF s(i,5)=s(i,10) THEN GO TO 3300
3285 IF i>103 THEN GO TO 3300
3290 NEXT e
3295 IF i=103 THEN GO TO 4500
3296 GO TO 3400
3300 INPUT "Amb aqueste no es pot<ENTER>";Q#
3310 INPUT "Tens un altre per jugar(s/n)";X#
3320 IF x#="s" THEN GO TO 2600
3330 IF x#="n" THEN GO TO 2000
3335 GO TO 3310
3400 REM Avancar fitxes
3410 IF s(i,4)=0 AND s(i,9)=0 THEN GO TO 4440
3420 IF s(i,11)<>7 THEN GO TO 4410
3430 IF s(i,5)=h OR s(i,10)=h THEN GO TO 4410
3500 REM "Matar" fitxes
3505 IF s(i,4)<>0 THEN GO TO 3540
3510 PRINT AT k(s(i,10),s(i,9),1),k(s(i,10),s(i,9),2); PAPER k(s(i,10),s(i,9),4)
;k(s(i,10),s(i,9),3)
3515 LET k(s(i,10),s(i,9),5)=0
3520 PRINT AT s(i,6),s(i,7); PAPER k(h,w,4);w
3530 LET s(i,10)=h: LET s(i,9)=w: LET k(h,w,5)=i

```



```

3535 GO TO 3580
3540 REM Matem fitxa de l'esquerra
3550 PRINT AT k(s(i,5),s(i,4),1),k(s(i,5),s(i,4),2); PAPER k(s(i,5),s(i,4),4);k(
s(i,5),s(i,4),3)
3555 LET k(s(i,5),s(i,4),5)=0
3560 PRINT AT s(i,1),s(i,2); PAPER k(h,w,4);w
3570 LET s(i,5)=h: LET s(i,4)=w: LET k(h,w,5)=i
3575 IF l=5 AND m=1 THEN LET m=0: GO TO 3650
3580 IF s(v,4)=w AND s(v,5)=h THEN GO TO 3620
3590 PRINT AT s(v,6),s(v,7); PAPER s(v,11);CHR# (s(v,8))
3600 LET s(v,10)=0: LET s(v,9)=0
3610 GO TO 3650
3620 PRINT AT s(v,1),s(v,2); PAPER s(v,11);CHR# (s(v,3))
3630 LET s(v,5)=0: LET s(v,4)=0
3650 REM Sumar 20
3660 LET l=20
3670 LET v=i
3680 GO TO 2600
4000 REM Control del sis
4005 LET c=0
4010 LET d=d+1
4020 IF d=3 THEN GO TO 4300
4030 FOR i=1 TO 4
4040 IF k(h,i,5)=0 THEN GO TO 4070
4050 NEXT i
4060 LET l=7
4070 FOR a=1 TO 4
4075 IF k(h,a,5)=0 THEN GO TO 4110
4080 FOR b=4 TO a+1 STEP -1
4090 IF k(h,b,5)=k(h,a,5) THEN GO TO 4130
4100 NEXT b
4110 NEXT a
4120 GO TO 4150
4130 IF k(h,a,5)>68 THEN GO TO 4100
4140 LET c=c+1: GO TO 4100
4150 IF c<1 THEN GO TO 2600
4155 LET c=c-1
4160 INPUT "Quina fitxa de la barrera ";w
4170 IF w<0 OR w>4 THEN GO TO 4160
4180 IF w=0 THEN GO TO 2000
4190 IF s(k(h,w,5),5)=s(k(h,w,5),10) THEN GO TO 2620
4200 GO TO 4160
4300 REM Mort per repeticio del sis
4310 IF i>68 THEN LET d=0: GO TO 2000
4320 PRINT AT k(h,w,1),k(h,w,2); PAPER k(h,w,4);k(h,w,3)
4330 IF s(i,4)=k(h,w,3) AND s(i,5)=h THEN GO TO 4360
4340 PRINT AT s(i,6),s(i,7); PAPER s(i,11);CHR# (s(i,8))
4350 LET s(i,9)=0: LET s(i,10)=0
4360 PRINT AT s(i,1),s(i,2); PAPER s(i,11);CHR# (s(i,3))
4370 LET s(i,4)=0: LET s(i,5)=0
4380 LET k(h,w,5)=0
4390 GO TO 2000
4400 REM Posar caracter
4410 IF s(i,4)=0 THEN GO TO 4440
4415 IF s(i,5)<>0 AND s(i,10) THEN GO TO 3300
4420 LET s(i,9)=w: LET s(i,10)=h: LET k(h,w,5)=i
4430 PRINT AT s(i,6),s(i,7); PAPER k(h,w,4);w
4435 GO TO 4460
4440 LET s(i,4)=w: LET s(i,5)=h: LET k(h,w,5)=i
4450 PRINT AT s(i,1),s(i,2); PAPER k(h,w,4);w
4460 IF s(v,4)=w AND s(v,5)=h THEN GO TO 4480
4470 PRINT AT s(v,6),s(v,7); PAPER s(v,11);CHR# (s(v,8)): LET s(v,9)=0: LET s(v,
10)=0: GO TO 2000
4480 PRINT AT s(v,1),s(v,2); PAPER s(v,11);CHR# (s(v,3)): LET s(v,4)=0: LET s(v,
5)=0: GO TO 2000
4500 REM Col.locacio d'arribada
4510 IF s(i+1,5)=h AND s(i+1,10)=h THEN GO TO 4570

```

```

4520 IF s(i+1,5)=0 THEN GO TO 4550
4530 PRINT AT s(i+1,6),s(i+1,7); PAPER s(i+1,11);w
4540 LET s(i+1,9)=w: LET s(i+1,10)=h: LET k(h,w,5)=i+1
4545 GO TO 4620
4550 PRINT AT s(i+1,1),s(i+1,2); PAPER s(i+1,11);w
4560 LET s(i+1,4)=w: LET s(i+1,5)=h: LET k(h,w,5)=i+1
4565 GO TO 4620
4570 IF s(i,10)<>0 THEN GO TO 4600
4580 PRINT AT s(i,6),s(i,7); PAPER s(i,11);w
4590 LET s(i,9)=w: LET s(i,10)=h: LET k(h,w,5)=i
4595 GO TO 4620
4600 PRINT AT s(i,1),s(i,2); PAPER s(i,11);w
4610 LET s(i,4)=w: LET s(i,5)=h: LET k(h,w,5)=i
4620 IF s(v,4)=w AND s(v,5)=h THEN GO TO 4630
4622 PRINT AT s(v,6),s(v,7); PAPER s(v,11);CHR$(s(v,8))
4624 LET s(v,10)=0: LET s(v,9)=0
4626 GO TO 4640
4630 PRINT AT s(v,1),s(v,2); PAPER s(v,11);CHR$(s(v,3))
4632 LET s(v,4)=0: LET s(v,5)=0
4640 IF s(i,5)<>0 THEN GO TO 4800
4660 REM Contar deu mes
4670 INPUT "Amb quina fitxa contes 10 ";w
4680 IF w<0 OR w>4 THEN GO TO 470
4690 IF w=0 THEN GO TO 2000
4700 LET l=10: GO TO 2620
4800 PRINT AT 10,10;"GUANYADOR": PRINT AT 12,14;"N. ";H
5000 DATA 0,11,1,1,11,2,2,11,3
5010 DATA 3,11,4,4,11,5,5,11,6
5020 DATA 6,11,7,7,11,8,7,9,9
5030 DATA 6,8,1,6,7,1,6,6,1
5040 DATA 6,5,1,6,4,1,6,3,1
5050 DATA 6,2,1,7,8,0,7,7,1
5055 DATA 7,6,2,7,5,3,7,4,4
5060 DATA 7,3,5,7,2,6,10,1,1
5070 DATA 11,1,7,14,2,1,15,2,8
5080 DATA 14,3,1,15,3,9,14,4,2
5090 DATA 15,4,0,14,5,2,15,5,1
5100 DATA 14,6,2,15,6,2,14,7,2
5110 DATA 15,7,3,14,8,2,15,8,4
5120 DATA 14,9,2,15,9,5,14,10,26
5130 DATA 15,10,27,16,10,28,17,10,29
5140 DATA 18,10,30,19,10,31,20,10,32
5150 DATA 21,10,33,21,18,35,20,18,36
5160 DATA 19,18,37,18,18,38,17,18,39
5170 DATA 16,18,40,15,18,41,14,18,42
5180 DATA 14,20,4,15,20,3,14,21,4
5190 DATA 15,21,4,14,22,4,15,22,5
5200 DATA 14,23,4,15,23,6,14,24,4
5210 DATA 15,24,7,14,25,4,15,25,8
5220 DATA 14,26,4,15,26,9,14,27,5
5230 DATA 15,27,0,10,28,5,11,28,1
5240 DATA 6,27,5,7,27,2,6,26,5
5250 DATA 7,26,3,6,25,5,7,25,4
5260 DATA 6,24,5,7,24,5,6,23,5
5270 DATA 7,23,6,6,22,5,7,22,7
5280 DATA 6,21,5,7,21,8,6,20,5
5290 DATA 7,20,9,7,18,60,6,18,61
5300 DATA 5,18,62,4,18,63,3,18,64
5310 DATA 2,18,65,1,18,66,0,18,67
6000 REM coordenades caselles
6002 DATA 0,12,145,0,0,0,13,146,0,0,7
6004 DATA 1,12,145,0,0,1,13,146,0,0,7
6006 DATA 2,12,145,0,0,2,13,146,0,0,7
6008 DATA 3,12,145,0,0,3,13,146,0,0,7
6010 DATA 4,12,145,0,0,4,13,146,0,0,6
6012 DATA 5,12,145,0,0,5,13,146,0,0,7
6014 DATA 6,12,145,0,0,6,13,146,0,0,7

```


6016 DATA 7,12,145,0,0,7,13,146,0,0,7
6018 DATA 8,9,147,0,0,9,9,148,0,0,7
6020 DATA 8,8,147,0,0,9,8,148,0,0,7
6022 DATA 8,7,147,0,0,9,7,148,0,0,7
6024 DATA 8,6,147,0,0,9,6,148,0,0,3
6026 DATA 8,5,147,0,0,9,5,148,0,0,7
6028 DATA 8,4,147,0,0,9,4,148,0,0,7
6030 DATA 8,3,147,0,0,9,3,148,0,0,7
6032 DATA 8,2,147,0,0,9,2,148,0,0,7
6034 DATA 10,2,147,0,0,11,2,148,0,0,3
6036 DATA 12,2,147,0,0,13,2,148,0,0,7
6038 DATA 12,3,147,0,0,13,3,148,0,0,7
6040 DATA 12,4,147,0,0,13,4,148,0,0,7
6042 DATA 12,5,147,0,0,13,5,148,0,0,7
6044 DATA 12,6,147,0,0,13,6,148,0,0,5
6046 DATA 12,7,147,0,0,13,7,148,0,0,7
6048 DATA 12,8,147,0,0,13,8,148,0,0,7
6050 DATA 12,9,147,0,0,13,9,148,0,0,7
6052 DATA 14,12,145,0,0,14,13,146,0,0,7
6054 DATA 15,12,145,0,0,15,13,146,0,0,7
6056 DATA 16,12,145,0,0,16,13,146,0,0,7
6058 DATA 17,12,145,0,0,17,13,146,0,0,3
6060 DATA 18,12,145,0,0,18,13,146,0,0,7
6062 DATA 19,12,145,0,0,19,13,146,0,0,7
6064 DATA 20,12,145,0,0,20,13,146,0,0,7
6066 DATA 21,12,145,0,0,21,13,146,0,0,7
6068 DATA 21,14,145,0,0,21,15,146,0,0,3
6070 DATA 21,16,145,0,0,21,17,146,0,0,7
6072 DATA 20,16,145,0,0,20,17,146,0,0,7
6074 DATA 19,16,145,0,0,19,17,146,0,0,7
6076 DATA 18,16,145,0,0,18,17,146,0,0,7
6078 DATA 17,16,145,0,0,17,17,146,0,0,2
6080 DATA 16,16,145,0,0,16,17,146,0,0,7
6082 DATA 15,16,145,0,0,15,17,146,0,0,7
6084 DATA 14,16,145,0,0,15,17,146,0,0,7
6086 DATA 12,20,147,0,0,13,20,148,0,0,7
6088 DATA 12,21,147,0,0,13,21,148,0,0,7
6090 DATA 12,22,147,0,0,13,22,148,0,0,7
6092 DATA 12,23,147,0,0,13,23,148,0,0,3
6094 DATA 12,24,147,0,0,13,24,148,0,0,7
6096 DATA 12,25,147,0,0,13,25,148,0,0,7
6098 DATA 12,26,147,0,0,13,26,148,0,0,7
6100 DATA 12,27,147,0,0,13,27,148,0,0,7
6102 DATA 10,27,147,0,0,11,27,148,0,0,3
6104 DATA 8,27,147,0,0,9,27,148,0,0,7
6106 DATA 8,26,147,0,0,9,26,148,0,0,7
6108 DATA 8,25,147,0,0,9,25,148,0,0,7
6110 DATA 8,24,147,0,0,9,24,148,0,0,7
6112 DATA 8,23,147,0,0,9,23,148,0,0,4
6114 DATA 8,22,147,0,0,9,22,148,0,0,7
6116 DATA 8,21,147,0,0,9,21,148,0,0,7
6118 DATA 8,20,147,0,0,9,20,148,0,0,7
6120 DATA 7,16,145,0,0,7,17,146,0,0,7
6122 DATA 6,16,145,0,0,6,17,146,0,0,7
6124 DATA 5,16,145,0,0,5,17,146,0,0,7
6126 DATA 4,16,145,0,0,4,17,146,0,0,3
6128 DATA 3,16,145,0,0,3,17,146,0,0,7
6130 DATA 2,16,145,0,0,2,17,146,0,0,7
6132 DATA 1,16,145,0,0,1,17,146,0,0,7
6134 DATA 0,16,145,0,0,0,17,146,0,0,7
6136 DATA 0,14,145,0,0,0,15,146,0,0,3
6200 REM coordenades caselles colors
6210 REM Caselles A
6212 DATA 1,14,145,0,0,1,15,146,0,0,6
6214 DATA 2,14,145,0,0,2,15,146,0,0,6
6216 DATA 3,14,145,0,0,3,15,146,0,0,6
6218 DATA 4,14,145,0,0,4,15,146,0,0,6

```
6220 DATA 5,14,145,0,0,5,15,146,0,0,6
6222 DATA 6,14,145,0,0,6,15,146,0,0,6
6224 DATA 7,14,145,0,0,7,15,146,0,0,6
6226 DATA 8,14,145,0,0,8,15,146,0,0,6
6228 DATA 9,14,145,0,0,9,15,146,0,0,6
6230 REM Caselles B
6232 DATA 10,3,147,0,0,11,3,148,0,0,5
6234 DATA 10,4,147,0,0,11,4,148,0,0,5
6236 DATA 10,5,147,0,0,11,5,148,0,0,5
6238 DATA 10,6,147,0,0,11,6,148,0,0,5
6240 DATA 10,7,147,0,0,11,7,148,0,0,5
6242 DATA 10,8,147,0,0,11,8,148,0,0,5
6244 DATA 10,9,147,0,0,11,9,148,0,0,5
6246 DATA 10,10,147,0,0,11,10,148,0,0,5
6248 DATA 10,11,147,0,0,11,11,148,0,0,5
6250 REM Caselles C
6252 DATA 20,14,146,0,0,20,15,146,0,0,2
6254 DATA 19,14,145,0,0,19,15,146,0,0,2
6256 DATA 18,14,145,0,0,18,15,146,0,0,2
6258 DATA 17,14,145,0,0,17,15,146,0,0,2
6260 DATA 16,14,145,0,0,16,15,146,0,0,2
6262 DATA 15,14,145,0,0,15,15,146,0,0,2
6264 DATA 14,14,145,0,0,14,15,146,0,0,2
6266 DATA 13,14,145,0,0,13,15,146,0,0,2
6268 DATA 12,14,145,0,0,12,15,146,0,0,2
6270 REM Caselles D
6272 DATA 10,26,147,0,0,11,26,148,0,0,4
6274 DATA 10,25,147,0,0,11,25,148,0,0,4
6276 DATA 10,24,147,0,0,11,24,148,0,0,4
6278 DATA 10,23,147,0,0,11,23,148,0,0,4
6280 DATA 10,22,147,0,0,11,22,148,0,0,4
6282 DATA 10,21,147,0,0,11,21,148,0,0,4
6284 DATA 10,20,147,0,0,11,20,148,0,0,4
6286 DATA 10,19,147,0,0,11,19,148,0,0,4
6288 DATA 10,18,147,0,0,11,18,148,0,0,4
6300 REM Coordenadas fitxas
6310 DATA 4,2,1,6,0
6320 DATA 4,3,2,6,0
6330 DATA 4,4,3,6,0
6340 DATA 4,5,4,6,0
6350 DATA 17,2,1,5,0
6360 DATA 17,3,2,5,0
6370 DATA 17,4,3,5,0
6380 DATA 17,5,4,5,0
6390 DATA 17,23,1,2,0
6400 DATA 17,24,2,2,0
6410 DATA 17,25,3,2,0
6420 DATA 17,26,4,2,0
6430 DATA 4,24,1,4,0
6435 DATA 4,25,2,4,0
6440 DATA 4,26,3,4,0
6450 DATA 4,27,4,4,0
9999 STOP
```


3 L'ordinador i les noves tecnologies informàtiques

4 Disseny i fabricació assistits per l'ordinador

18 Estructura d'un ordinador

32 Disseny i fabricació de circuits integrats

41 Informàtica a l'escola

45 Recull de basic

57 Altres programes. El joc del parxís

Col·lecció

connecta el micro

CURS DE BASIC

- 1 FEM INFORMÀTICA
- 2 IMATGES I SONS
- 3 TRACTAMENT DE LA INFORMACIÓ
- 4 ROBOTS I INTEL·LIGENCIA ARTIFICIAL
- 5 **NOVES TECNOLOGIES**

Servei de Publicacions de la



FUNDACIÓ CAIXA DE PENSIONS

El programa «Connecta el micro,
pica l'start» és una iniciativa
conjunta de Caixa de Pensions
«la Caixa» i TV3

275 Ptes.



<https://electrodreams.blog/>