

“Un ordenador é un sistema dixital síncrono que realiza, dun xeito moi eficiente, tarefas programadas”

Nesta definición hai termos que precisan concreción.

Sistema: Dispositivo que reacciona ante unhas **entradas** ofrecendo os resultados a través dunhas **saídas**. Normalmente un usuario humano é o destinatario desta comunicación, pero pode ser outro dispositivo.

Dixital: O funcionamento, as entradas e mais as saídas están nun formato discreto (non continuo) en **binario** (con símbolos que toman unicamente valores 0 e 1 que se chaman **bits**). Toda a información manexada polo ordenador é neste formato. Así por exemplo, segundo o uso que faga del, o seguinte **byte** (secuencia de 8 bits) 01110010 pode ser o número enteiro 114, o número real 6,025, a letra “r”, ou a orde para que o computador faga unha operación de suma. Se o destinatario é humano, o formato debe traducirse para poder ser lexible. A **palabra** é a unidade de información, e ten un tamaño fixo en cada ordenador.

Síncrono: Os pulsos dun sinal de reloxo regulan o ritmo de funcionamento do sistema dixital. A frecuencia do dito reloxo é da orde de 10^9 pulsos por segundo.

Eficiente: O ordenador é capaz de executar nun segundo máis de 1000 millóns de instrucións. Por exemplo, é quen de facer nese tempo billóns de divisións de números con 20 cifras decimais. Se puxésemos de acordo a toda a poboación mundial para facer simultaneamente operacións de división teriamos que facer 130 contas nun segundo para manter o ritmo dun ordenador normalíño, e sen un só erro.

Tarefas programadas: Secuencia de accións básicas, **instrucións**, que realiza o ordenador unha tras outra e na orde en que están escritas. Estas secuencias denomínanse **programas**, e o conxunto de programas chámase **software**. Recorda que as instrucións usan formato binario.



Fig. 1

FIG 1. Nun computador os dispositivos de comunicación (entrada/saída) son o máis evidente para o usuario. A súa función é actuar como interface co usuario. Constitúen un dos catro compoñentes funcionais dun computador definidas polo modelo Von Neumann.

“Os elementos que compoñen un ordenador son: a unidade de control, a calculadora, a memoria e os dispositivos de entrada/saída”

Von Neumann definiu en 1945 como debería ser e funcionar un computador dixital, e o seu esquema segue agora vixente. Ten habido intentos de definir modelos diferentes pero non tiveron éxito e foron descartados. O modelo definido por Von Neumann consta de catro elementos básicos: unha calculadora, a unidade de control, a unidade de almacenaxe ou memoria, e os dispositivos de entrada/saída para a comunicación co exterior. A maior parte dos elementos electrónicos deste sistema atópanse integrados nun chip, o **microprocesador**. Ademais estableceu a relación que hai entre eles, e como se conectan entre si, así como o seu funcionamento segundo os conceptos de **programa almacenado** (as instrucións gárdanse na mesma memoria que os datos) e de **ciclo de execución** das instrucións.

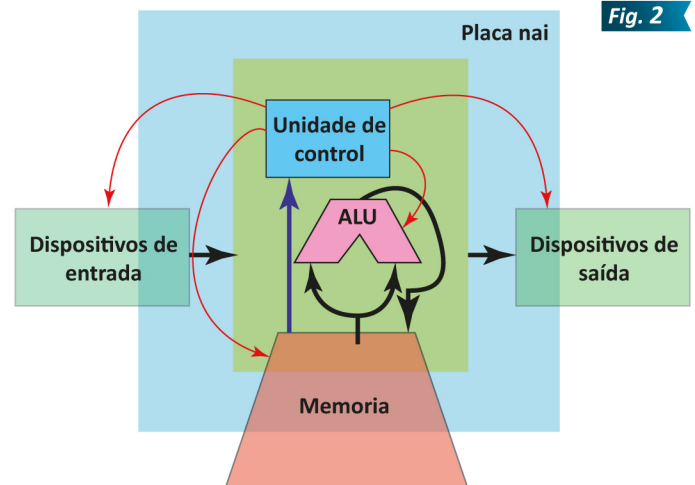


Fig. 2

FIG 2. A arquitectura Von Neumann define os compoñentes hardware dun computador e as súas conexións. A calculadora ou ALU realiza operacións aritméticas simples, a memoria almacena datos e instrucións, os dispositivos de entrada/saída serven para as comunicacións co exterior, e a unidade de control activa o funcionamento de todos os demais elementos a través de liñas de control (frechas vermellas) despois de ler na memoria unha instrución (frecha azul). Os datos necesarios para realizar operacións na ALU lense da memoria (frechas negras) e do exterior a través dos dispositivos de entrada. Os resultados ofrécense ao exterior a través dos dispositivos de saída. A unidade de control e a ALU constitúen o que se chama procesador ou CPU, e se están nun único microchip chámase microprocesador. Habitualmente un microprocesador tamén inclúe unha pequena parte da memoria (registros e memoria caché). O microprocesador está conectado nunha placa de circuíto impreso chamada placa nai, onde se establecen as conexións con outros chips que inclúen parte da memoria (RAM e caché) e as controladoras dos dispositivos de entrada/saída. Finalmente unha parte importante da memoria atópase fóra da placa nai. Esta é a memoria secundaria e o exemplo típico é o disco duro.

“A unidade de control descodifica unha instrución en cada ciclo de reloxo e activa as liñas de control necesarias para que se leve a cabo”

As **liñas de control** son xeradas pola unidade de control e van ao resto do sistema: á calculadora, por exemplo, para indicar cales son a operación aritmética e os operandos; á memoria se hai que ler ou modificar o valor almacenado nalgunha posición de memoria concreta; ou aos dispositivos de entrada/saída, se se require algunha acción de comunicación co exterior.

“A unidade aritmetolóxica realiza operacións aritméticas simples moi rapidamente”

A **ALU** (“Arithmetic and Logic Unit”) é a calculadora contida no microprocesador. Está, polo tanto, fabricada coa mellor tecnoloxía electrónica do momento, polo que é rapidísima. Pero non ten moitas operacións integradas: o usual é que soamente suma, reste, multiplique e divida, ademais de facer operacións moi básicas con bits: OR, AND, desprazamentos, etc. Algunhas tamén son quen de calcular raíces cadradas. Non precisa máis. As operacións e funcións máis sofisticadas están programadas, son software. Os operandos e o resultado almacénanse en **rexistros** próximos á ALU.

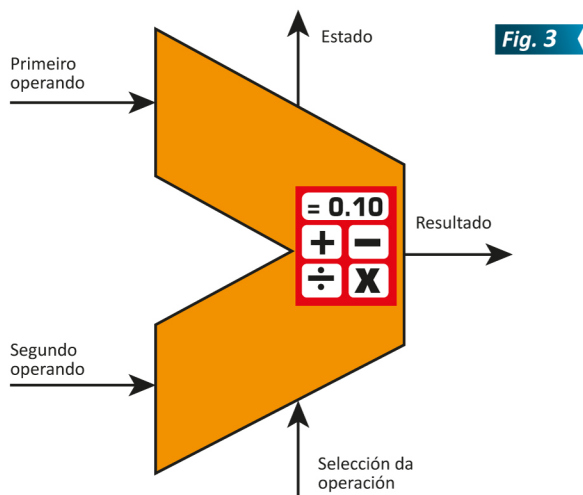


FIG 3. A figura é unha representación típica dunha ALU como elemento de circuío. Precisa dúas entradas para ler os operandos e unha saída para o resultado. O número de bits dos datos é finito e coincide co tamaño dos rexistros do procesador, tipicamente soen ser 32 ou 64 bits. A este tamaño denomínase **palabra**, e é decidida polo fabricante do microprocesador. A ALU necesita liñas de control dende a unidade de control para seleccionar a operación que vaia realizar, e ofrece sinais de estado para indicar posibles incidencias, por exemplo que o resultado sexa tan grande que non se poida representar cos bits das palabras (situación que se chama “overflow”).

Pódense ter dúbidas sobre se unha calculadora tan rápida é necesaria, cando a maioría dos programas non son eminentemente computacionais, e non requiren moitos cálculos. Pero debe terse en conta que en cada unha das instrucións, sexa cal sexa, tense que facer polo menos unha suma para calcular a posición de memoria da seguinte instrución. Moitas veces esta operación é sinxela, simplemente sumar unha unidade ao valor da posición da instrución actual.

Cando o programa require facer un gran número de operacións aritméticas pode delegar o labor nun **hardware** específico que recibe o nome xenérico de **coprocesador**. En particular as **tarxetas gráficas** enmárcanse nesta definición e están especificamente deseñadas para traballar con vectores ou matrices, e polo tanto son axeitadas para o procesamento de imaxes, aínda que tamén poden ser utilizadas para outras computacións.

“A memoria está organizada xerarquicamente, e a información cópiase duns niveis a outros”

A visión da memoria para os programadores é unha secuencia de posicións consecutivas. O número de bits necesario para identificar estas posicións soe coincidir co tamaño da **palabra**. O conxunto de posicións de memoria posibles chámase **memoria virtual**. Esta secuencia de posicións debe conter todas as instrucións do código dun programa e todos os datos cos que traballa. Os programadores fan referencias a estas posicións de memoria para identificar onde están os datos e as instrucións, e así acceder a eles. Esta visión idealista é moi sinxela pero afastada do **hardware** real da memoria.

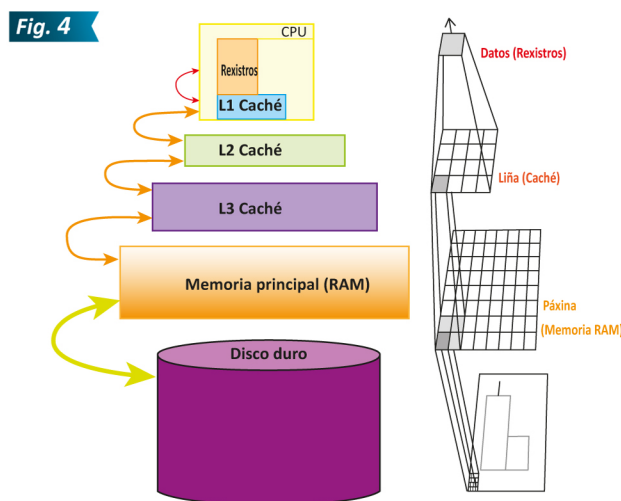


FIG 4. A memoria dun computador está fisicamente organizada nunha xerarquía. Cando o procesador solicita o acceso a unha determinada posición, primeiro búscase nos rexistros, se non está neles búscase na caché do primeiro nivel, e así sucesivamente ata chegar ao nivel do disco duro. A medida que se baixa de nivel, os tempos de acceso incrementáanse enormemente, polo que convén ter a información o máis alto posible na xerarquía. Para conseguir isto, cando un dato se move entre un nivel e o seguinte faíno conxuntamente cos datos próximos na memoria denominados páxinas no caso da memoria principal (frecha verde na figura) e liñas no das memorias caché (frechas vermellas). Un detalle adicional é que a información non se move entre niveis, cópiase, de xeito que tipicamente hai unha copia dun dato en todos os niveis de memoria dende o nivel máis alto no que se atopa.

Fig. 5

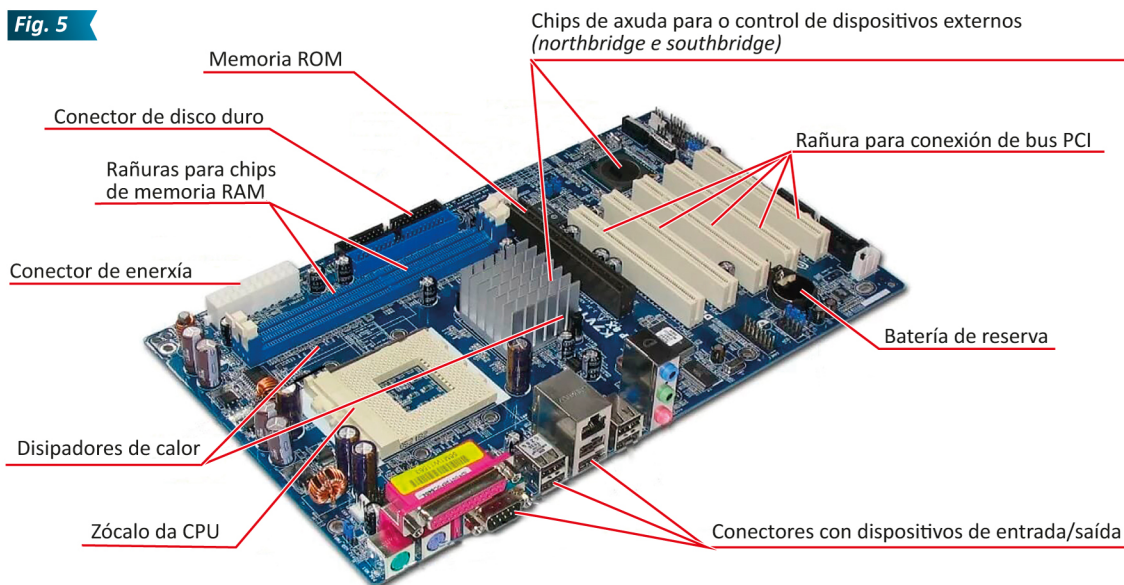


FIG 5. A placa nai é un circuío impreso no que se conectan dispositivos electrónicos e chips que inclúen o microprocesador, parte da memoria, as controladoras dos dispositivos de entrada/saída, chips e memorias de arranque do sistema coa súa propia pequena batería, etc. A frecuencia de reloxo á que traballan os chips actuais é tan elevada que unha boa parte da superficie da tarxeta ten que ser protexida con disipadores de calor. Para a conexión cos dispositivos de entrada/saída, a placa nai dispón de conectores con diferentes estándares de comunicación específicos de cada tipo de dispositivo.

A memoria adoita estar fisicamente organizada nunha estrutura xerarquizada en niveis, os niveis máis altos corresponden a memorias especialmente rápidas e pequenas denominadas **cachés**, de xeito que un sistema pode ter dous ou tres niveis de caché, as do primeiro nivel son especialmente rápidas e pequenas, e adoitan estar integradas no microprocesador. Os seguintes niveis de caché son algo máis lentos e grandes e soen estar en chips próximos ao procesador. Por debaixo das cachés na xerarquía está a **memoria RAM** tamén chamada **memoria principal** ou **memoria física** que é moito máis grande que as cachés pero máis lenta (unha orde de magnitude aproximadamente). Fisicamente son chips conectados ao procesador na **placa nai**. O nivel máis baixo da xerarquía é o **disco duro** que é especialmente lento e de gran capacidade.

Durante a execución dun programa, cando o procesador require o acceso a un dato ou a unha instrución, debe identificarse o nivel no que fisicamente se atopa. Esta identificación chámase **tradución de memoria virtual a física** e é responsabilidade do sistema operativo coa axuda de *hardware* específico (a **MMU**: unidade de manexo da memoria) que está integrada no propio microprocesador. Esta estrutura funciona marabillosamente porque en todos os programas se cumpre o **principio de localidade** que di que na execución dun programa as posicións de memoria requiridas en tempos próximos están en posicións de memoria próximas. Deste xeito cando o procesador solicita un dato que está no disco, cópiase na RAM xunto cos datos próximos (este conxunto chámase **páxina**), xa que o procesador no futuro próximo con alta probabilidade precisará datos próximos (da mesma páxina) que xa estarán na RAM. O mesmo principio funciona para a copia de datos dende a RAM á caché e entre os diferentes niveis de caché. Neste caso o grupo de datos consecutivos copiados son moitos menos e chámanse **liñas**.

“Os dispositivos de entrada/saída son xestionados polo sistema operativo”

Caracterízanse por un funcionamento moi lento debido a que moitas veces son mecánicos ou eléctricos. A súa funcionalidade é basicamente a de establecer o intercambio de información entre o procesador e o usuario ou outra máquina. Os códigos que os manexan forman parte do sistema operativo, e chámanse **controladores** ou **drivers**, e os dispositivos *hardware* que xestionan o seu funcionamento particular denomínanse **controladoras** (en feminino). As ordes e o movemento de información entre as controladoras e o procesador almacénanse en posicións específicas das controladoras chamadas **portos**. Transmítense a través de condutores eléctricos que constitúen os **buses de comunicación**. A conexión entre a placa nai e os dispositivos de entrada/saída establécese con diferentes protocolos e conexións en función do tipo de información e da velocidade requirida.

Como os dispositivos de entrada/saída son moi lentos comparados coa velocidade do procesador, o sistema operativo bloquea momentaneamente os programas implicados nelas para atender outros que eventualmente estean esperando para seguir a súa execución. Este mecanismo de cambio de programa en execución chámase **cambio de contexto**. As **interrupcións** son mecanismos que permiten que os dispositivos de entrada/saída avisen ao sistema de que algunha operación de entrada/saída rematou ou dalgunha outra incidencia exterior. Tamén o sistema operativo é responsable da súa xestión.

“O funcionamento dun ordenador consiste na execución dunha instrución en cada pulso de reloxo”

En cada ciclo de reloxo realízase, é dicir, execútase, unha instrución. Para executala Von Neumann estableceu unha serie de

pasos que se deben realizar un tras outro de xeito ordenado. A esta secuencia de pasos chámase **ciclo de execución das instrucións**. Son os seguintes:

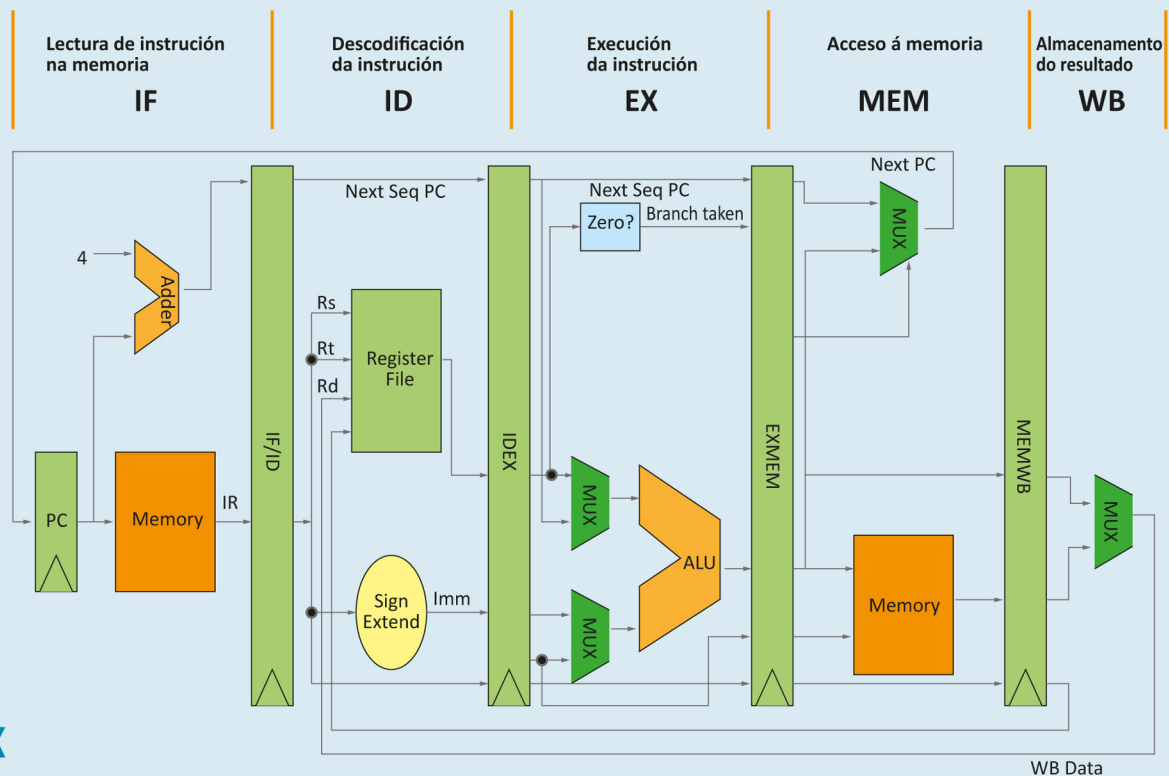


Fig. 6

FIG 6. O ciclo de execución das instrucións está establecido na arquitectura Von Neumann e consta de cinco fases. Na primeira lese da memoria a instrución que se vaia executar. A posición de memoria está nun rexistro do microprocesador denominado contador de programa (PC na figura) e a instrución lida almacénase noutro rexistro denominado rexistro de instrucións (IF/ID na figura). Na segunda fase traballa a unidade de control para activar os elementos hardware asociados á execución da instrución. Na terceira fase realízanse as accións necesarias para executar a instrución, por exemplo unha operación da ALU. Na cuarta fase accédese á memoria para ler ou escribir datos se a instrución o require. Na derradeira fase escríbese o resultado da instrución nun rexistro do microprocesador. En cada ciclo de reloxo execútanse estas fases ininterrompidamente dende a primeira ata a última instrución do programa.

1. Ler a instrución da memoria. Para isto debe coñecerse a súa posición ou enderezo virtual. Esa posición está gardada nun rexistro especial chamado **contador de programa**. Unha vez lida, almacénase noutro rexistro especial chamado **rexistro de instrucións**.
2. Descodificar a instrución. A instrución almacenada no rexistro de instrucións interprétase, é dicir, a partir dos valores dos bits da instrución actívanse as liñas de control necesarias ao resto do *hardware* para que faga o que está indicado na instrución. Iso faíno a unidade de control.
3. Executar a instrución. Activando os elementos *hardware* responsables da acción programada.
4. Acceder á memoria para ler ou escribir un dato nunha posición determinada.
5. Completar a instrución. No **rexistro de estado** almacénase información relativa a como foron as cousas na execución da correspondente acción. Calcúlase a posición virtual en memoria onde está a seguinte instrución en ser executada e se a acción asociada á instrución actual finaliza con algún resultado, almacénase no rexistro correspondente.

A execución deste ciclo repítese indefinidamente ata que o programa finaliza. En realidade, a execución nestas cinco etapas faise de xeito segmentado, é dicir, nun **pipeline**: mentres unha instrución está na fase 1, a previa está na fase 2, a anterior na 3... No seguinte ciclo, a que estaba na fase 1 pasa á fase 2, a que estaba na 2 pasa á 3, e así sucesivamente. Deste xeito mellórase a velocidade de execución do mesmo xeito que unha cadea de produción nunha fábrica.

O conxunto de instrucións de cada ordenador forma parte xunto co *hardware* do que se denomina **arquitectura do ordenador**. As instrucións pódense agrupar atendendo á súa funcionalidade nuns poucos tipos: instrucións de lectura dun dato na memoria para levalo a un rexistro do procesador, instrucións de escritura na memoria para copiar un dato dun rexistro ata unha posición determinada da memoria, instrucións aritméticas que realizan algunha operación na calculadora con operandos almacenados en rexistros, e finalmente operacións de fluxo de execución que permiten determinar saltos no programa para que a seguinte instrución en ser executada non sexa a que está na seguinte posición de memoria, que é o comportamento por defecto.

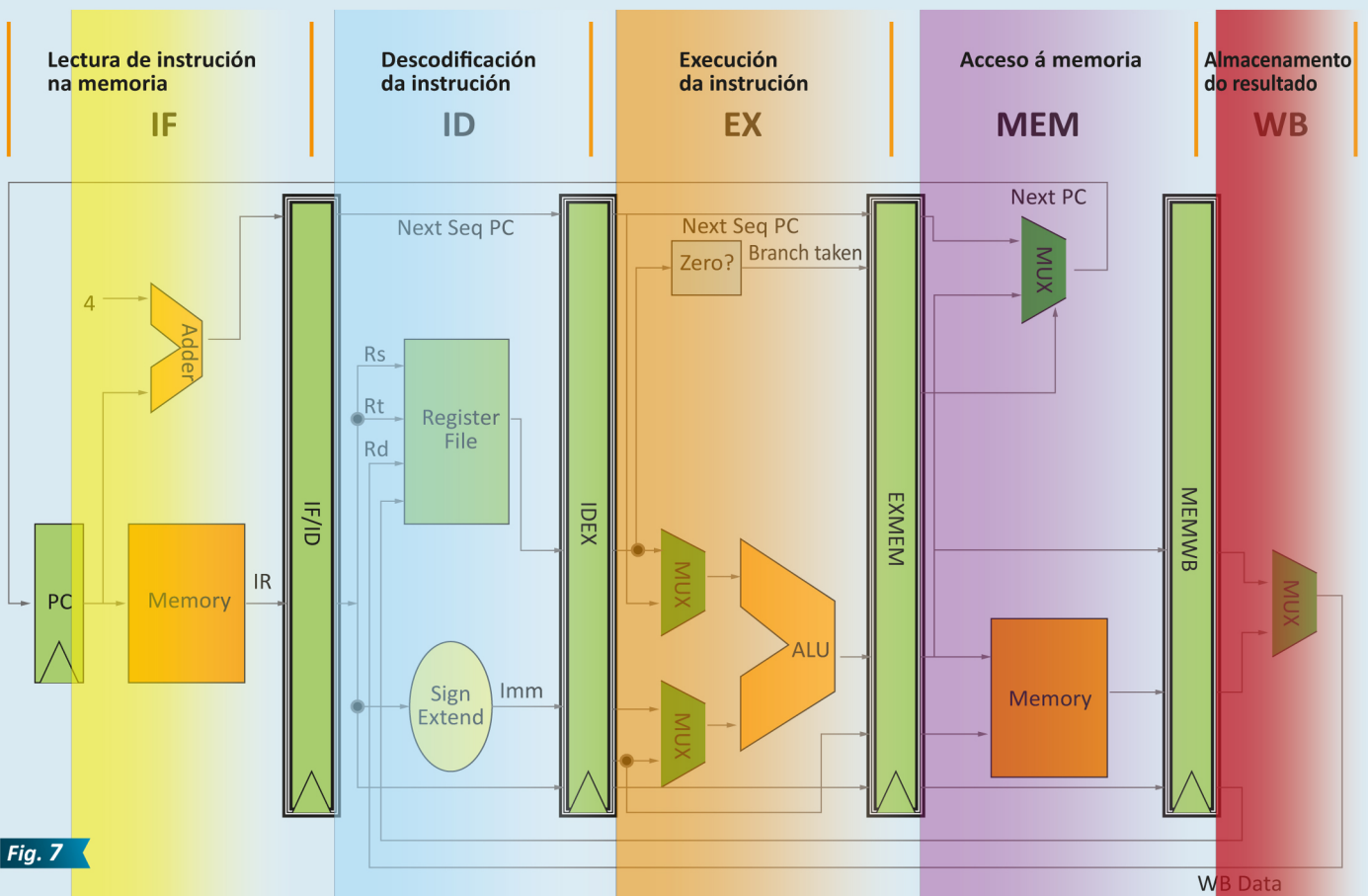


Fig. 7

FIG 7. Nos procesadores apróveitase a execución en fases das instrucións para facelo como unha cadea de produción. Esta estratexia denomínase “pipeline”, de xeito que todas as fases están traballando simultaneamente, pero cada unha cunha instrución diferente. Son precisos rexistros específicos que separen as fases para que o resultado dunha delas poida ser usada como entrada da seguinte no seguinte ciclo de reloxo. Na figura móstrase un momento dado o estado das fases, cada cor está asociada a unha instrución diferente, de xeito que hai cinco instrucións activas, cada unha delas ocupando unha etapa diferente da execución. De xeito que no ciclo seguinte cada unha delas ocupará a seguinte etapa, a última terá rematado, e incorporárase unha nova na primeira.

“Unha secuencia ordenada de instrucións constitúe un programa, que cando se executa pasa a ser un proceso”

Un programa é un conxunto ordenado de instrucións binarias. Os programadores non teñen que programar usando este formato (**linguaxe máquina**) senón que programan empregando **linguaxes**

de alto nivel máis intelixibles para o ser humano, usando palabras clave en inglés como “if”, “then”, “while”, “go to”, “otherwise”, “print”, ... e símbolos matemáticos: “+”, “/”, “*”, ...

Ao conxunto de programas denomínase **software**. E hai programas de aplicación útiles para o ser humano e a súa comunicación:

xogos, navegadores, simuladores de fenómenos físicos, editores de textos, ferramentas para debuxar, correo electrónico, contornas de fiestras, etc. Outros códigos serven para a xestión do *hardware* e do resto de programas: chámase **software de base** e inclúe o sistema operativo, os compiladores, os *drivers* dos dispositivos de entrada/saída...

Un programa en execución é un **proceso**. Xa dende os primeiros ordenadores se aplicou o concepto de **multiprogramación**: nun momento dado existen varios procesos activos, pero soamente un deles está realmente executándose e o resto esperan a súa quenda para seguir avanzando. Para o ser humano dá a impresión de que todos os programas van avanzando concorrentemente, cando en realidade o avance faise en pequenos intervalos de tempo de uso do procesador que son repartidos equilibradamente entre os programas en execución. Eses intervalos chámanse **quantum** e o sistema operativo é o encargado de decidir tras cada *quantum* cal é o seguinte programa en adquirir o uso do procesador. Cada *quantum* ten unha duración duns poucos milisegundos, polo que o usuario non é consciente deste reparto microscópico de uso da CPU entre os procesos.

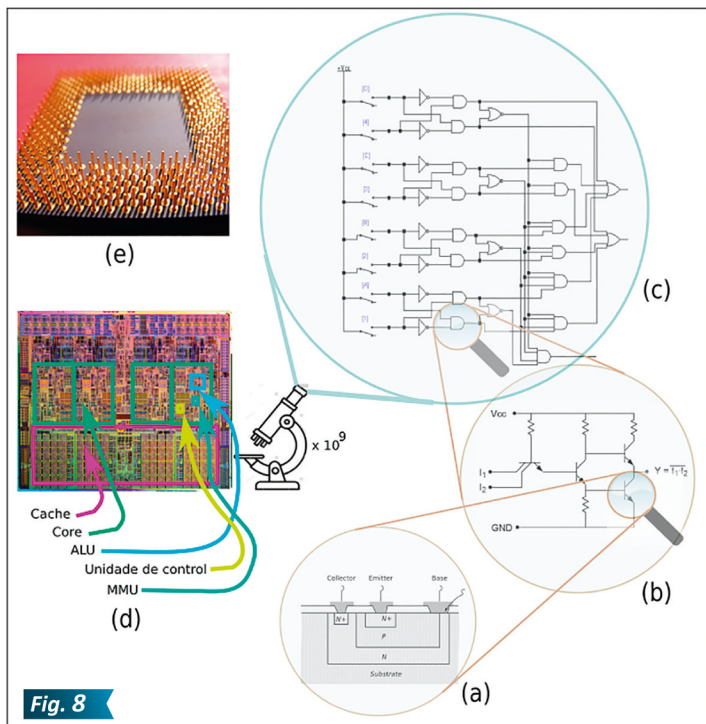


Fig. 8

FIG 8. Os dispositivos electrónicos fabricanse con tecnoloxía electrónica baseada no concepto de transistor que é un elemento conmutador controlado por tensión (figura a). A presenza ou ausencia de tensión eléctrica establece o valor do bit en cada posición do circuito. Combinando axeitadamente varios transistores pódense deseñar portas lóxicas (figura b). Conectando diversas portas lóxicas pódense definir circuitos dixitais que poden realizar operacións aritméticoalóxicas complexas en binario (figura c), usando para iso a álgebra de Boole. Con estes circuitos pódense construír calculadoras, unidades de control, memorias, etc. Os microprocesadores son circuitos moi sofisticados que inclúen moitos destes elementos nun espazo moi pequeno (varias decenas de millóns de transistores por centímetro cadrado) (figura d). Para incorporalos a unha placa nai temos que encapsulalos con tantos pins de conexión como sexan necesarios (normalmente centenaes) (figura e).

“Os programadores escriben programas usando linguaxes de alto nivel que deben ser traducidas a linguaxe máquina por un compilador”

Un bo programador é aquel que escribe os seus programas dun xeito claro e **modular** (con partes claramente identificables para as distintas tarefas) e seguindo as recomendacións e directrices consensuadas, de xeito que outros programadores poidan colaborar na programación entendendo sen esforzo os módulos xa pro-

gramados e o seu uso. Existe outro aspecto para ter en conta, a programación eficiente, con alto rendemento (fundamentalmente alta velocidade) que consiste en escribir códigos pensando en como se van executar no ordenador, para o cal é preciso ter un coñecemento detallado dos compoñentes *hardware* e do *software* de base. Aínda que un programador normalmente non vai programar en linguaxe máquina para aproveitar a arquitectura do sistema, pero pode escribir un programa nunha linguaxe de alto nivel tratando de aproveitar a tradución que fará o **compilador**.

O compilador é un programa que traduce un código escrito nunha linguaxe de alto nivel a unha versión en linguaxe máquina. Agora os compiladores ofrecen resultados da tradución moi eficientes e dificilmente superables incluso para un programador experto en linguaxe máquina.

```
program fibonacci
implicit nome
integer i, nmax
integer s, a, aux
nmax = 10 !Poner un limite
s = 1: a = 1 !Inicializar
!Iniciar loop
WRITE (*,*)
do = i = 1, nmax
!Imprimir elemento i de la serie
WRITE (*,*) 'Elemento', i, 'de la serie de Fibonacci:',
WRITE (*,*)
!Calcular nuevo elemento de la serie
aux = s
s = s + a
a = aux
end do
end program fibonacci
```

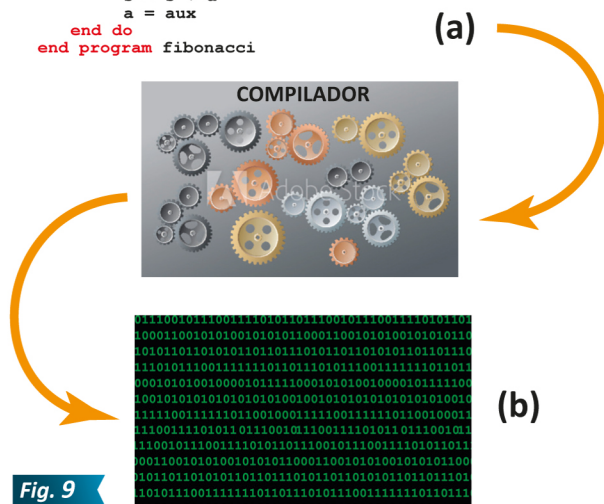


Fig. 9

FIG 9. Os programadores usan linguaxes de alto nivel (próximas á comprensión do ser humano) para escribiren os códigos (figura a). Estes códigos son textos que usan mnemotécnicos e sintaxes facilmente entendibles polos humanos. É preciso traducir os programas en secuencias de instrucións escritas en linguaxe máquina, é dicir, binario (figura b). Iso fano programas especializados denominados compiladores. O principal requisito dun compilador é que a tradución sexa semanticamente correcta, é dicir, que cando se execute o programa faga o que espera o usuario. O segundo requisito é que o resultado sexa eficiente, ou sexa, que cando se execute o programa, o faga moi rápido. Os compiladores son tan efectivos que moitas veces incluso para o programador é difícil recoñecer o seu programa orixinal na tradución.

“O sistema operativo é un programa que xestiona o funcionamento global do sistema”

O sistema operativo é un programa que ten dúas funcionalidades: facilitar a vida dos usuarios (por exemplo a través dunha contorna de fiestras) e xestionar o funcionamento a baixo nivel do ordenador. No contexto desta segunda funcionalidade, hai tres tarefas particularmente importantes:

- Realizar **chamadas ao sistema**: son saltos nos programas a códigos que xestionan a actividade do sistema operativo. Estes códigos constitúen o que se chama **kernel do sistema operativo**. Nel accédese a partes críticas do *hardware* ou da memoria ás que o usuario non debe ter acceso directo por motivos de seguridade e de integridade do sistema. É como pedirlle ao sistema operativo que faga algo por nós cando o que queremos facer é delicado ou perigoso para a integridade do sistema.

- Xestionar **interrupcións**. As interrupcións son avisos que chegan do exterior para chamar a atención do computador, por exemplo, mover o rato avisa ao ordenador que ten que reaccionar dalgún xeito, esa reacción está programada no sistema operativo.
- **Traducir** as posicións de memoria solicitadas pola unidade de control, chamadas posicións virtuais, en posición de memoria concretas na xerarquía de memoria: cachés, RAM, disco, denominadas posicións físicas.

“Para mellorar o rendemento os microprocesadores inclúen máis dun procesador”

Actualmente os procesadores presentan varios **núcleos** ou **cores**, é dicir, están compostos por varias unidades funcionais independentes que son quen de executar un proceso cada unha delas simultaneamente. Son os **sistemas multinúcleo**. Este tipo de *hardware* non invalida a vixencia da arquitectura Von Neumann, simplemente faina máis ampla duplicando parte do hardware.

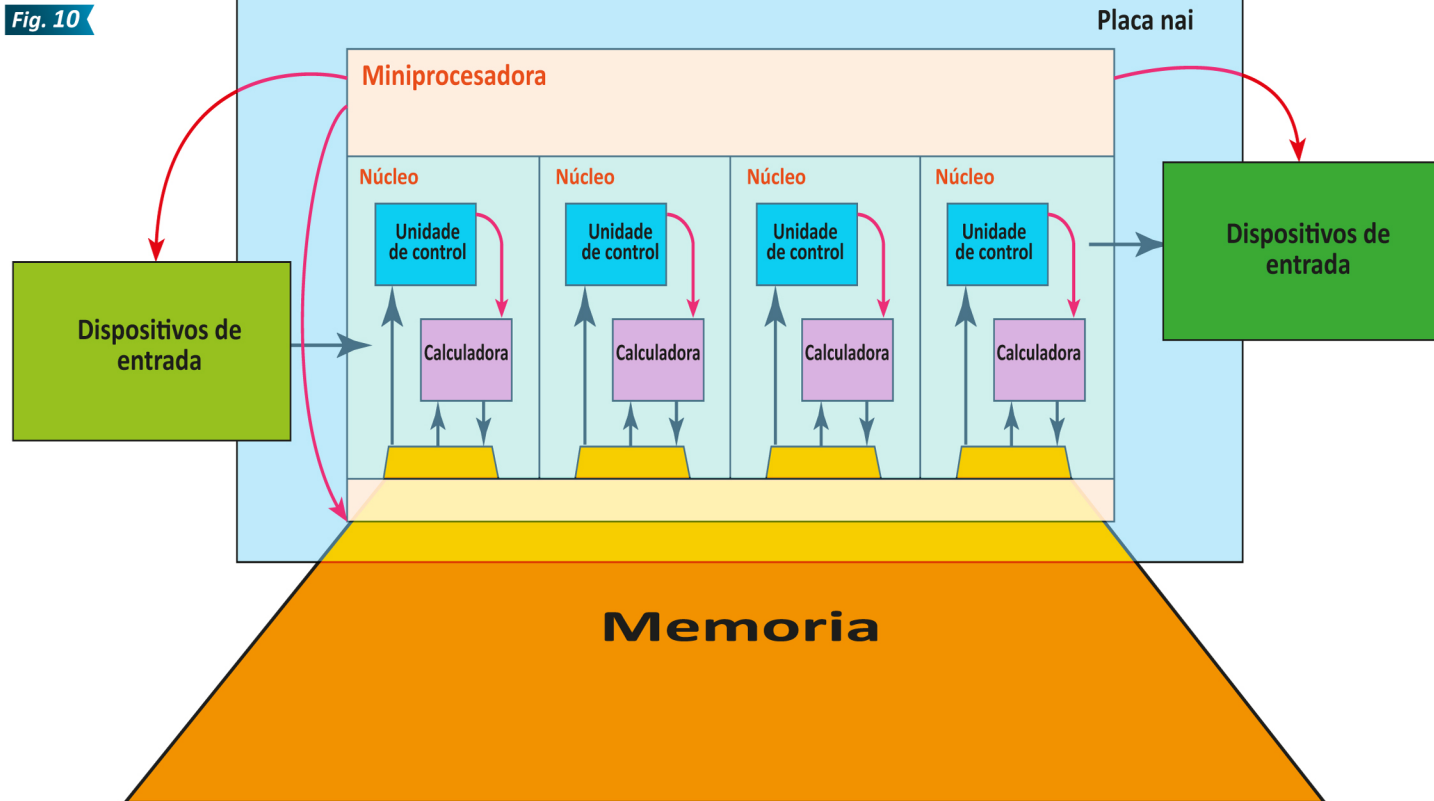


FIG 10. O nivel de integración na fabricación de chips chegou a tal punto que no mesmo chip caben varios procesadores. A estes procesadores chámaseslles núcleos ou cores. O seu funcionamento segue rexido pola arquitectura Von Neumann, salvo que cada un executa un proceso diferente. Como se pode ver na figura os niveis máis altos de memoria caché están repartidos entre os núcleos. Os niveis baixos da xerarquía de memoria son compartidos. É un reto actualmente a programación eficiente deste tipo de sistemas, baseado nun modelo de programación chamado programación paralela, e que é significativamente máis complexo que o tradicional secuencial.

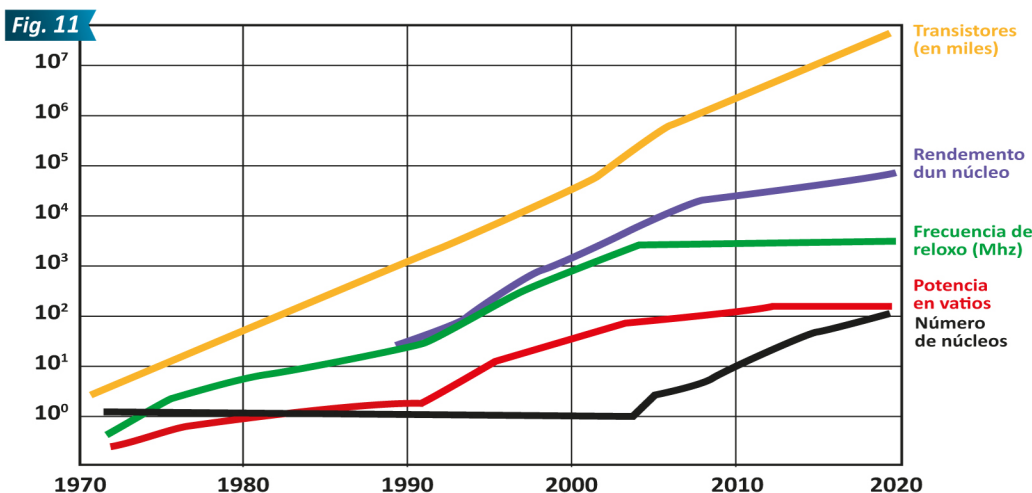


FIG 11: Evolución dos microrprocesadores nos últimos 50 anos. As principais conclusións destes datos son: o número de transistores que poden fabricarse nun chip aumenta exponencialmente, duplicase cada dous anos. A velocidade das CPUs non crece ao mesmo ritmo nos últimos 15 anos, o que fai que os fabricantes aumenten o número de núcleos. A frecuencia de reloxo estancouse dende hai uns 20 anos por problemas tecnolóxicos no funcionamento dos transistores e pola enerxía consumida a maior frecuencia.

