

## El sistema básico de entrada/salida (BIOS)

### Sumario

<b>23.1. Tipos de BIOS</b> . . . . .	<b>6</b>
23.1.1. Actualizables: Flash BIOS . . . . .	7
23.1.2. Autoconfiguración: PnP BIOS . . . . .	10
23.1.2.1. El hardware . . . . .	10
23.1.2.2. El firmware . . . . .	11
23.1.2.3. El software . . . . .	12
<b>23.2. Elementos ligados a la BIOS</b> . . . . .	<b>13</b>
23.2.1. La RAM CMOS . . . . .	13
23.2.1.1. Concepto . . . . .	13
23.2.1.2. Características . . . . .	13
23.2.1.3. Etimología . . . . .	14
23.2.1.4. Evolución . . . . .	14
23.2.1.5. Contenidos . . . . .	15
23.2.1.6. Implementaciones . . . . .	16
23.2.2. La pila . . . . .	16
23.2.3. Borrado de la RAM CMOS . . . . .	17
<b>23.3. Funciones de la BIOS</b> . . . . .	<b>18</b>
23.3.1. Suministrar el sistema básico de entrada/salida . . . . .	18
23.3.1.1. Tratamiento de interrupciones hardware y software . . . . .	20
23.3.1.2. Ejemplo: Actualización del reloj del sistema . . . . .	22
23.3.2. Iniciación del sistema . . . . .	25
<b>Resumen</b> . . . . .	<b>29</b>
<b>La anécdota: Flash, esa pequeña gran protagonista</b> . . . . .	<b>30</b>
<b>Cuestionario de evaluación</b> . . . . .	<b>31</b>

El hardware y el software del computador interactúan a través de un interfaz, consistente en un conjunto de rutinas en código máquina encargadas de llevar a cabo las tareas de más bajo nivel del computador, esto es, aquellas que más dependen del hardware de nuestro equipo. Estas rutinas se almacenan en una memoria no volátil que forma parte integrante de la placa base del computador y que se conoce popularmente con el nombre de **BIOS** (*Basic Input Output System - Sistema Básico de Entrada/Salida*).

BIOS

pág. 18

firmware

Como veremos en la [sección 23.3.1](#), su denominación proviene de la función que le dió su razón de ser. La BIOS puede considerarse a la vez como hardware (físicamente podemos tocarla) y como software (contiene un compendio de rutinas), por lo que se clasifica en una capa intermedia del computador que se sitúa entre ambas y que se conoce como **firmware**.

Precisamente es esta estratégica ubicación de la BIOS la que le ha permitido desacoplar las capas software y hardware del computador, algo que ha contribuido de forma decisiva al éxito del PC desde su nacimiento. Si alguna vez nos hemos preguntado cómo se consigue que los programas que se ejecutaban en los primeros PC puedan aún ser ejecutados en plataformas contemporáneas cuyos microprocesadores, buses, memorias y placas base son radicalmente diferentes, la respuesta está en proporcionar a la capa software una interfaz estándar e independiente del dispositivo que permita al hardware evolucionar sin que el software se perciba de ello.

independencia  
del dispositivo**Ejemplo 23.1: EL PAPEL DE LA BIOS ANTE LOS CAMBIOS DEL HARDWARE**

Un programa que quiera realizar una operación de escritura en disco duro, solicitará la ejecución de la rutina de la BIOS encargada de efectuar esa operación. Esa rutina es la única que conoce las formas de diálogo con el controlador del disco duro en términos de sincronización, temporización y control. Cuando sustituyamos el disco duro por otro más rápido, es suficiente con notificar a la BIOS la nueva secuencia de temporización para que cuando el programa solicite esa misma operación, ésta pueda llevarse a cabo sobre el nuevo dispositivo y aproveche sus mejores prestaciones.

Si el hardware en lugar de modificarse se amplía, entonces la BIOS se amplía en la misma línea para dar cobertura a los servicios que incorpora el nuevo dispositivo. En ambos casos, el *driver* o *controlador software* del dispositivo utilizará estas nuevas funciones, pero los servicios antiguos no se ven alterados, con objeto de que puedan seguir siendo utilizados por nuestros viejos programas.

De su dualidad software-hardware, de su ubicación a tan bajo nivel, y de la función tan ingrata que se le ha asignado, podemos imaginar que la BIOS no es un componente fácil de describir. Afortunadamente, la vertiente hardware sólo admite las variantes de implementación que describiremos en la [sección 23.1](#). Con excepción de ésta y de la sección dedicada a la RAM-CMOS y la pila, el resto del capítulo versa sobre la faceta software de la BIOS.

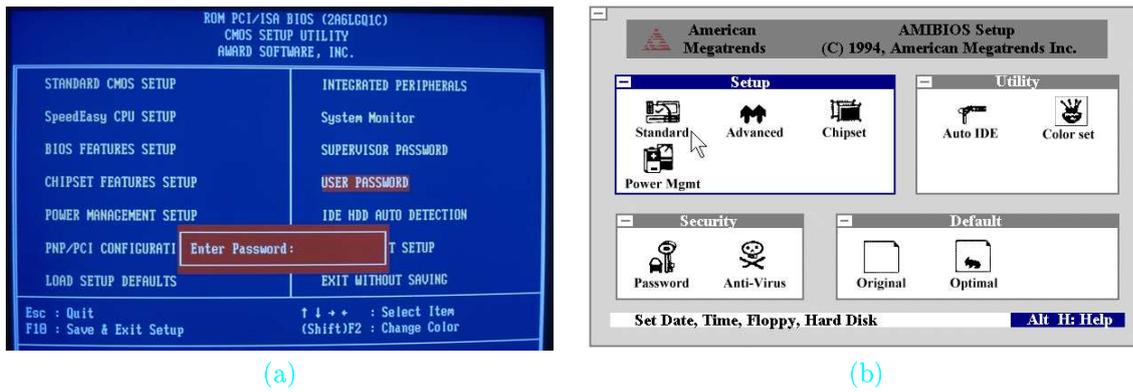
**SECCIÓN 23.1****Tipos de BIOS**

de dispositivos

En función de lo que ya sabemos, es fácil imaginar que cada PC tiene una versión de BIOS diferente atendiendo a las características de su hardware. Lo que resulta más chocante es que en realidad existen varias BIOS dentro del PC, cada una especializada en uno de los dispositivos periféricos de que dispone el sistema. Por ejemplo, la tarjeta gráfica dispone de su BIOS para albergar instrucciones implementadas de forma cableada que visualizan la información gráfica, y de forma similar operan los discos duros y los controladores SCSI.

del sistema

No obstante, se ha acuñado el término BIOS para referirse de forma genérica a la BIOS del sistema, es decir, a la que cubre los *servicios básicos de entrada/salida* de las aplicaciones. Esta convención también ha sido adoptada por nosotros aquí: Habida cuenta de que nuestra obra se



(a)

(b)

FOTO 23.1: Aspecto que presenta el menú principal de la BIOS de dos de los fabricantes líderes del mercado. (a) Una BIOS de Award, más sobria pero respaldada por mejor documentación. (b) Una BIOS de AMI, visualmente más atractiva por su entorno de ventanas e iconos.

encarga mucho antes del sistema en su conjunto que del tratamiento de los periféricos en particular, para nosotros no habrá otra BIOS que no sea la del sistema en general.

Unos pocos fabricantes de placas base construyen sus propias BIOS en función de las necesidades de los sistemas a los que destinan sus productos, mientras que otros prefieren encargárselas a fabricantes de BIOS proporcionándoles una especificación completa sobre estas necesidades. A medida que la placa base evoluciona, los chips BIOS evolucionan de su mano para dar servicio eficiente a las mejoras hardware introducidas por los fabricantes.

Los tres principales fabricantes de BIOS son Microid Research, AMI (American Megatrends Inc), Award Computer y Phoenix Technologies. Decimos tres y no cuatro porque Phoenix adquirió Award allá por septiembre de 1998 y desde entonces son la misma compañía aunque siga manteniéndose la dualidad de marcas. Award es el líder de ventas y el que suministra las BIOS mejor documentadas, mientras que AMI incorpora un interfaz más coqueto con entorno de ventanas y manejo desde ratón (ver [foto 23.1](#)).

principales  
fabricantes

La marca en sí no es algo a lo que debemos prestar especial atención, ya que no es garantía de nada: Todas las marcas tienen BIOS con prestaciones pobres y otras de gama alta cuyas características son más recomendables. La [foto 23.2](#) muestra un amplio abanico de chip BIOS de diferentes fabricantes.

importancia  
de la marca

← pág. 8

Además, la marca no la elegimos nosotros, sino el fabricante de la placa base que decidamos adquirir. Lo que sí podemos hacer es elegir un modelo que dentro de esa marca nos proporcione unas buenas prestaciones, ya que si la BIOS no está preparada para entenderse con un nuevo dispositivo que sí soporta la placa base, trasladaremos la responsabilidad de la configuración a la capa software del sistema operativo. En ese caso, será el correspondiente controlador software del dispositivo el encargado de dejar residente en algún área de la memoria principal todas las rutinas que debieron estar incluidas en la BIOS. Por ello, debemos asegurarnos de que las ampliaciones que podamos acometer el día de mañana sobre nuestro equipo van a tener la cobertura correspondiente en forma de actualización de la BIOS (ver [sección 23.1.1](#)), o de que el sistema va a ser capaz de reconfigurarse automáticamente para trabajar de forma adecuada con cada nueva incorporación (ver [sección 23.1.2](#)).

previsión  
futura

← pág. 10

## Actualizables: Flash BIOS

### ◀ 1.1

Para que la memoria que forma parte de la BIOS no perdiese sus contenidos cuando apagamos el equipo, inicialmente se implementó mediante una memoria ROM (Read Only Memory) de sólo

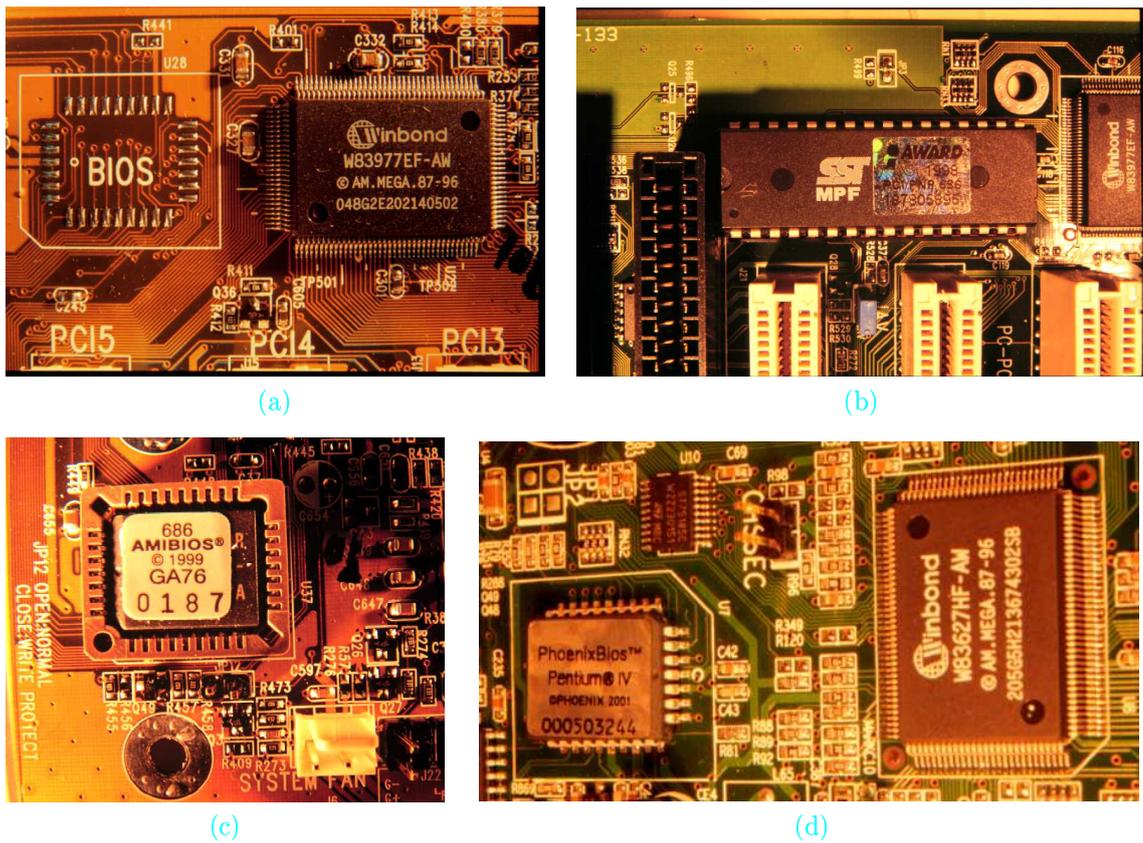


FOTO 23.2: Diversos modelos comerciales de chip BIOS: (a) Modelo de 1996 de American Megatrends Inc. (AMI) para quinta generación sobre una placa base dotada de habitáculo de conexión adicional para otro chip en previsión de futuras averías y eventual recambio. (b) Modelo de 1998 de Award para sexta generación (Pentium III) en formato DIP desmontable. (c) Modelo de 1999 de AMI para séptima generación (K7) sobre un zócalo que permite la fácil sustitución del chip. (d) Modelo de 2001 de Phoenix para séptima generación (Pentium 4), que consta del mismo chip de quinta generación como versión base (a la derecha), y un conjunto de servicios ampliados como los servicios de bajo consumo o monitorización de variables eléctricas (a la izquierda).

**ROM-BIOS** lectura y contenido invariable (de ahí procede su acepción más popular, **ROM-BIOS**).

Posteriormente, se facilitó la actualización de sus valores mediante algún procedimiento externo, antecediendo la letra P de *Programmable* al nombre, y existiendo tres variantes que enumeramos a continuación de menos a más recomendable:

- UV-PROM** ① **UV-PROM (Ultra-Violet PROM)**. Sólo puede actualizarse mediante un procesamiento dirigido por rayos ultravioleta (UV-PROM). Es la alternativa menos interesante porque requiere de una instrumentación especializada que difícilmente va a estar al alcance del usuario doméstico. La única operación fácil de realizar es su borrado, que puede llevarse a cabo prescindiendo de una etiqueta adhesiva reflectante que lleva en su lomo, y que previene de la entrada de luz procedente del exterior.
- EEPROM** ② **EEPROM (Electrically Erasable PROM)**. Se trata de una memoria programable y borrrable electrónicamente, cuyos contenidos pueden modificarse mediante algún procedimiento electrónico bastante más sencillo de llevar a cabo que la variante anterior.

- ③ **Flash.** Las memorias de tipo Flash son un subconjunto de las EPROM que permiten su actualización tantas veces como se desee mediante un mecanismo tan simple como la aplicación de una **tensión superior**. El chip tiene, por lo tanto, tres regímenes de trabajo:
- a) Sin alimentación. Retiene sus contenidos, diferenciándose de la RAM tradicional.
  - b) A 3.3 ó 5 voltios, se comporta como una memoria ROM convencional, permitiendo únicamente operaciones de lectura.
  - c) A 12 voltios, se transforma en una memoria RAM de sólo escritura cuyos contenidos pueden ser escritos totalmente o en una serie de palabras indicando simplemente su dirección. Esta escritura selectiva es su rasgo más distintivo frente a una EEPROM convencional.

Flash  
voltaje:  
- 0v.  
- 5v.  
- 12v.

Los valores de tensión están elegidos convenientemente por ser los más usuales en los cables de corriente procedentes de la fuente de alimentación (ver [tabla 32.7](#)), como los cables rojo (5 voltios) y amarillo (12 voltios) que alimentan a los chips y los motores de los discos, respectivamente. Sin embargo, frente a éstos, que tienen **intensidades de corriente** superiores a 10 amperios que aumentan progresivamente con la potencia de la fuente de alimentación, las líneas de alimentación que conducen a la Flash-BIOS se alimentan de los cables blanco y azul por los que fluyen 0.5 amperios con independencia de la fuerza de la fuente de alimentación. Para completar la información referente a la memoria Flash, remitimos al lector a la [sección 13.7.1](#).

☛ Vol.5 en Web  
corriente  
☛ p.172/Vol.2

La presencia de una BIOS de tipo Flash en nuestro equipo suele distinguirse por venir acompañada de dos elementos inseparables:

- ① Un **jumper JAV** situado junto a la BIOS encargado de regular la conmutación al modo de escritura en el chip. Este jumper se aprecia en la [foto 23.2.b](#) (en color azul, parte central inferior de la foto, un poco más abajo del chip BIOS de Award).

jumper JAV  
☛ pág. 8

Si no encontramos el jumper, puede ser porque la conmutación al modo de escritura no se regula mediante un elemento hardware como el *jumper*, sino mediante un programa que directamente utiliza una interrupción para invocar a la rutina de la BIOS que realiza tal función como aditivo (alternativa software), o bien directamente mediante una opción del menú **BIOS FEATURES SETUP** de la BIOS (alternativa firmware - ver la opción **FLASH WRITE PROTECT** dentro de la [sección 24.3.3](#)).

☛ pág. 45

#### ⚠ Riesgo 23.1: LA POSICIÓN POR DEFECTO DE LOS JUMPERS

En el año 2002 comenzó a instalarse en el mercado de placas base la costumbre de entregar desde fábrica el producto con el jumper JAV colocado en la posición de escritura en la Flash-BIOS.

Dada la extremada fragilidad del chip BIOS cuando se encuentra en esta posición, muchos PC impiden arrancar el sistema con el jumper así colocado. Por lo tanto, una vez finalizado el montaje del equipo, comprobaremos la posición de este jumper antes de proceder a su encendido.

En otros modelos de placas base, el sistema tampoco arranca si el jumper JCC se encuentra colocado en la posición que habilita el borrado de la RAM-CMOS (ver [sección 23.2.3](#)).

**disquete**      ② Un disquete suministrado por el fabricante de la placa base que nos proporciona las sucesivas versiones actualizadas junto con las instrucciones para grabar éstas en nuestro chip BIOS y el programa encargado de llevar a cabo estas operaciones. En caso de no disponer de este disco, tampoco pasa nada, ya que la gran mayoría de fabricantes publican en su página Web de Internet cualquier tipo de actualización que necesitemos (consultar los enlaces <http> que proporcionamos en la bibliografía). Este método tiene el atractivo de asegurarnos estar al día con las novedades que vayan saliendo al mercado en el futuro.

pág. 11 ➔

La **foto 23.3** muestra el aspecto de una BIOS de tipo Flash junto al jumper JAV que regula su escritura y la serigrafía en la placa base que nos alerta sobre su significado.

vulnerabilidad

pág. 188 ➔

Ahora bien, la comodidad de uso de un sistema suele estar reñida con su seguridad. En el caso de la Flash-BIOS, existe la posibilidad de que esta facilidad para modificar la BIOS pueda ser aprovechada por ciertos virus informáticos. Para prevenir y reparar este tipo de desaguisados, consultar la **sección 28.6.1**.

usos

En general, y a pesar de esta vulnerabilidad, merece la pena adquirir una Flash-BIOS. Las variantes de actualización de nuestro sistema en el futuro son difíciles de predecir, y a menudo la velocidad en la evolución de la tecnología nos obliga a ampliar el equipo antes de lo deseado. El caso más usual es la renovación del microprocesador: Por ejemplo, el cambio de K7 Thunderbird a Athlon XP es uno de los muchos que requiere una actualización de la BIOS.

proliferación

La enorme utilidad de la Flash-BIOS ha tenido un claro reflejo en el mercado, donde su **proliferación** ha sido tal que resulta complicado encontrar otro tipo de tecnología para la BIOS. La anécdota del final del capítulo proporciona una serie de datos económicos que corroboran esta percepción.

## 1.2 ▶ Autoconfiguración: PnP BIOS

PnP

La especificación *Plug & Play* (*Enchufa y Juega*), mnemotécnicamente conocida como PnP, es la capacidad de un sistema para: (a) reconocer cualquier tipo de dispositivo que incorporemos al mismo, y (b) realizar los ajustes necesarios de configuración para que pueda utilizarse sin ningún tipo de conflicto.

diálogos estándar

*Plug & Play* no es más que un conjunto de diálogos estándar, un protocolo común en el que todos los elementos del sistema pueden preguntarse los aspectos internos de su organización, pasarse esta información, y proceder a la definición de una configuración conjunta para el sistema que lo haga funcionar de forma completamente automática.

requisitos

Para que un sistema PnP desempeñe adecuadamente la función de autoconfiguración las tres capas del sistema deben implementar la especificación *Plug & Play*: El hardware, el firmware y el software.

### 1.2.1 El hardware

nuevo

Un dispositivo que aspire a poder autoconfigurarse debe tener implementados en su controlador los diálogos para recibir la petición de información acerca de sus parámetros internos y enviar los valores correspondientes. Tanto los módulos de memoria como las diversas tarjetas AGP, PCI, ISA o EIDE incorporan esta valiosa característica desde finales de los años 90.

antiguo

Si sus componentes son más antiguos y no incluyen las prestaciones *Plug & Play*, esto no significa que no puedan funcionar en un sistema que sí lo sea; simplemente, estos elementos no van a entender los diálogos PnP y tendrán que ser configurados manualmente mediante alguna alternativa (jumpers, menú en BIOS, ...).

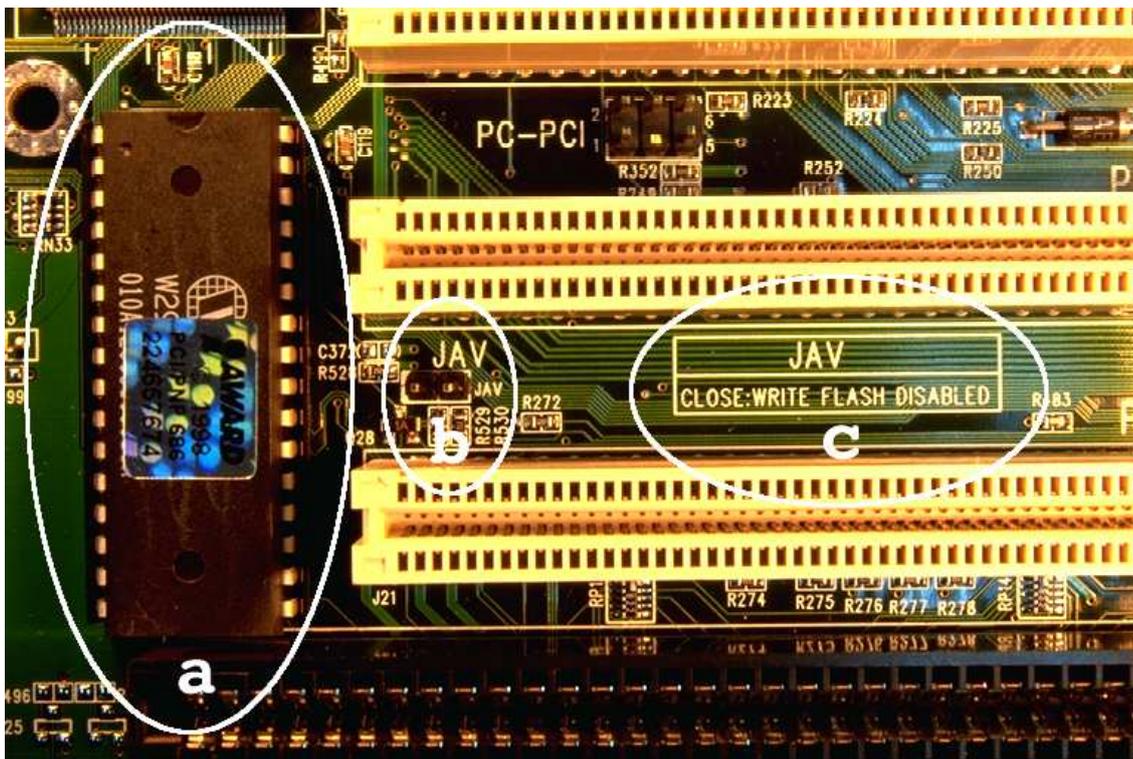


FOTO 23.3: (a) Aspecto de una BIOS de tipo Flash. En concreto, se trata de un modelo de Award Computer muy común en placas base para Pentium III. (b) El jumper JAV responsable de controlar la actualización de la BIOS cuando ésta es de tipo Flash. (c) Serigrafía de la placa base que explica el significado de dicho jumper: Cuando está puesto, desactiva la escritura del chip, con lo que habrá que quitarlo cuando queramos actualizar la BIOS.

### 1.2.2 El firmware

Una BIOS que implemente la especificación PnP debe ser capaz de identificar cada uno de los dispositivos conectados al sistema y realizar su configuración automática. Estas tareas se llevan a cabo siguiendo la secuencia de pasos que enumeramos a continuación:

- |   |                             |
|---|-----------------------------|
| <p>❶ Identificar los recursos que los dispositivos PnP necesitan para trabajar, como líneas de interrupción que van a ser asignadas a la ocurrencia de eventos externos en estos dispositivos, direcciones de memoria que van a actuar como puertos de comunicaciones, y líneas de DMA por las que el dispositivo pueda acceder a memoria de forma directa.</p> | <p>identificar recursos</p> |
| <p>❷ Recopilar un mapa inicial provisional de asignación de líneas de interrupción, puertos y líneas de DMA a cada dispositivo.</p>   | <p>asignar líneas</p>       |
| <p>❸ Rastrear las ROM de estos dispositivos para extraer sus características hardware.</p>  | <p>rastrear ROM</p>         |
| <p>❹ Habilitar los periféricos e informarles de los recursos que les han sido asignados.</p>  | <p>habilitar disps.</p>     |

La especificación PnP para la BIOS fue desarrollada conjuntamente por Compaq, Intel y Phoenix Technologies. La [foto 23.4](#) muestra un ejemplar de este tipo. Para que una BIOS sea compatible con esta especificación, debe soportar 13 llamadas adicionales a funciones del sistema que establecen las formas de diálogo con un sistema operativo PnP. Estas llamadas se incorporaron en

← [pág. 12](#)

génesis

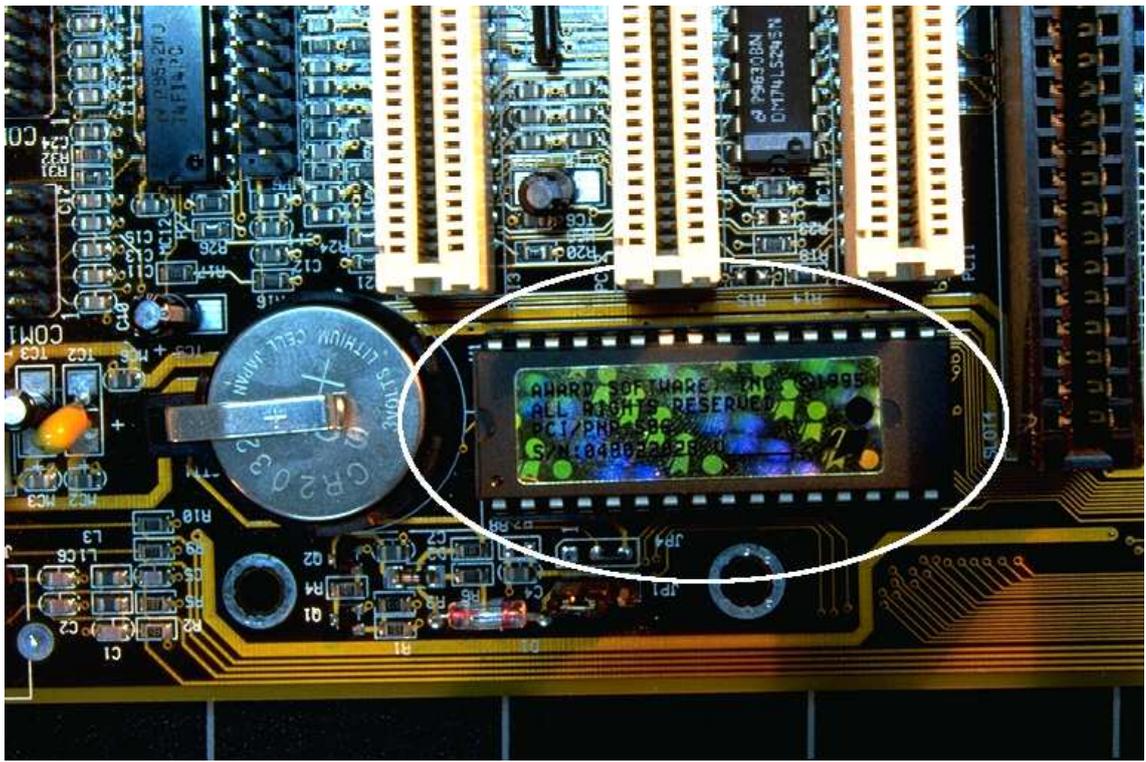


FOTO 23.4: Una BIOS de tipo PnP fabricada por Award Computer junto a la pila del sistema.

las BIOS a partir de mediados de 1995, fecha en la que quedó establecido el estándar PnP definitivo. Si nuestra BIOS es del tipo Flash, aunque sea de una fecha anterior, siempre podremos actualizarla para que recoja la especificación PnP.

### 1.2.3 El software

Una vez que los dispositivos han comunicado al sistema los recursos que necesitan para trabajar, la BIOS y el sistema operativo trabajan conjuntamente para resolver los posibles conflictos entre dispositivos e informarles de los recursos específicos que finalmente podrán utilizar.

conflictos

De existir algún conflicto irresoluble, corresponde al sistema operativo informar al usuario de esta circunstancia. Dependiendo de la sofisticación del sistema operativo, el usuario podrá entonces usar el propio sistema operativo para configurar el dispositivo conflictivo de forma manual, o bien apagar el sistema y manipular de forma directa los conmutadores de su tarjeta para cambiar la configuración.

Cuando se reinicia el PC, el sistema operativo comprueba de nuevo los posibles conflictos, informando al usuario de las posibles eventualidades surgidas tras la nueva configuración. El proceso se repite hasta que todos los dispositivos presentan una configuración compatible.

disponibilidad

Prácticamente todos los sistemas operativos actuales (Windows y las últimas versiones de DOS, OS/2 y Linux) incluyen la especificación PnP, por lo que ninguno de los tres pilares vistos (hardware, firmware y software) presentan problemas desde el punto de vista de su disponibilidad PnP.

## Elementos ligados a la BIOS

### La RAM CMOS

### ◀ 2.1

#### 2.1.1 Concepto

Prácticamente desde sus orígenes, los computadores se han cimentado en los postulados de la denominada “arquitectura Von Neumann”. Uno de esos postulados propugna que las instrucciones y los datos de que se compone un programa se representan y almacenan conjuntamente en la memoria del computador.

Para diferenciar un tipo de información de otro, cualquier programa ejecutable se encuentra estructurado desde entonces en partes o segmentos diferenciados por su contenido, siendo los dos segmentos más importantes el segmento de código, que contiene las instrucciones, y el segmento de datos, que contiene los valores con los que éstas realizan sus operaciones.

segmentos

En base a esto, lo que la BIOS almacena es el segmento de código de un compendio de programas o rutinas de bajo nivel. Cabe preguntarse entonces quién almacena el segmento de datos para estos programas.

Por ejemplo, si la BIOS facilita la selección de la frecuencia de funcionamiento del procesador por parte del usuario suministrándole un coqueto interfaz (ver [sección 24.3.2](#)), luego deberá registrar la selección del usuario en algún lugar. Muchos usuarios creen que estos valores se almacenan en la propia BIOS, pero a poco que reflexionemos nos daremos cuenta de que esta opción es inviable, puesto que el carácter permanente de sus contenidos obliga a acometer su actualización mediante procedimientos más sofisticados que una mera instrucción `sw`<sup>1</sup>.

pág. 43

La memoria que albergue los datos que necesitan los programas de la BIOS requerirá así una **implementación** muy sutil. Los primeros niveles de la jerarquía de memoria del PC, como el banco de registros, las memorias caché o la memoria principal, no pueden recepcionar esta información, pues en todos los casos sus contenidos desaparecen cuando apagamos el equipo, y esto obligaría a introducir todos los valores a cada nuevo encendido de la máquina. Tampoco sirve el último nivel de la jerarquía constituido por la memoria de disco, pues aunque el soporte magnético presente este anhelado carácter permanente, su extremada lentitud es a todas luces inadecuada para una función de tan bajo nivel. Por todo ello, es necesario habilitar una nueva forma de memoria que de cobertura de forma exclusiva a los datos que necesitan los programas de la BIOS. Esta memoria se denomina **RAM-CMOS**.

requerimientos

concepto

#### 2.1.2 Características

Las **características** de la RAM-CMOS pueden resumirse en los siguientes cuatro aspectos:

- ❶ Un **tamaño discreto**, que suele oscilar entre los 64 y los 256 bytes.
- ❷ Una **alimentación independiente**, lo que le permite retener sus contenidos cuando el equipo se encuentra apagado. Esta alimentación se lleva a cabo desde una pila o un acumulador (ver [sección 23.2.2](#)).

pequeña

no volátil

pág. 16

<sup>1</sup>Según el nemotécnico más frecuente en lenguaje ensamblador, *save word*.

- RAM      ③ **Acceso independiente** a cualquiera de sus posiciones proporcionando la dirección correspondiente (memoria RAM).
- CMOS      ④ Una **tecnología de integración CMOS** (Complementary Metal Oxide Semiconductor), que por su reducido consumo permite prolongar al máximo la duración de la pila.

### 2.1.3 Etimología

nomenclatura

Como puede apreciarse, el nombre de RAM-CMOS proviene de las dos últimas características listadas anteriormente, aunque éstas distan mucho de ser propiedad exclusiva de este chip. En primer lugar, la memoria de tipo RAM es la más abundante del sistema, y en segundo lugar, la tecnología CMOS es la más utilizada para fabricar los chips, incluyendo los propios módulos de memoria principal.

En nuestra opinión, el nombre es bastante desacertado, pero lo cierto es que a ninguna otra forma de memoria del sistema se le menciona explícitamente con el nombre de RAM-CMOS aunque también lo sea, y como tampoco se trata de ir contracorriente en algo que ya se encuentra bastante establecido, adoptaremos esta misma nomenclatura a partir de ahora.

### 2.1.4 Evolución

Al contrario que la BIOS, la RAM-CMOS no formó parte integrante del PC desde su nacimiento, o al menos no en la forma en la que ahora la conocemos.

Inicialmente, la mayoría de los contenidos que ahora almacena la RAM-CMOS eran fijos y se encontraban alojados en diversas memorias ROM diseminadas a lo largo y ancho de la placa base.

conmutador DIP

pág. 36

Posteriormente, algunos fabricantes fueron desgajando de ellos los parámetros de configuración más dinámicos y situándolos en unos conmutadores DIP (Dual In-Line Package) que se encontraban dispersos en la geografía de la placa base y que podían ser manipulados externamente para cambiar a otras configuraciones preestablecidas. La [foto 24.1](#) muestra uno de estos conmutadores. Sus principales inconvenientes eran la necesidad de abrir el PC para poder acceder a ellos y la dificultad para establecer una correspondencia semántica entre la posición de cada conmutador y su efecto asociado.

conversión a memoria

MC146818

p. 13/Vol. 3

La evolución tecnológica ha provocado un aumento del área de datos requerido por la BIOS. Esta circunstancia, junto a la consabida evolución de las placas base hacia diseños de espacio más reducido, propiciaron la sustitución de los conmutadores DIP por memoria RAM-CMOS, solución que permitía como valor añadido una fácil y cómoda ampliación del espectro de parámetros configurables, algo nada desdeñable dada la vorágine de cambios en que se vio inmerso el PC a partir de entonces. Los primeros chips que se vieron de este tipo fueron los **146818 de Motorola**, ya presentes desde el IBM-AT en formato DIP de 24 pines, y que incluían conjuntamente en su interior el reloj de tiempo real (RTC - ver [sección 17.2.4](#)).

De esta manera, el chip de memoria podía actualizarse por software: Primero, desde un programa incluido en el disco de diagnóstico del equipo, y más tarde, desde un programa de agradable aspecto y cómodo manejo que podía ser ejecutado durante la secuencia de arranque del sistema y cuyo código se incorporó a la ROM-BIOS. Había nacido el programa de configuración del sistema o Setup, que será objeto de un detallado estudio en el [capítulo 24](#).

Setup

pág. 35

actualización de sus valores

Posteriormente, también se permitió la actualización de algunos de estos parámetros directamente desde el sistema operativo, como la fecha y hora del sistema, los colores iniciales de la pantalla, o los parámetros que actualizan los drivers de los dispositivos.

Todos estos pasos fueron llevándose a cabo de forma gradual por los distintos fabricantes, conviviendo durante algún tiempo el almacenamiento de ciertos parámetros en la RAM-CMOS

con la selección manual de otros, donde los conmutadores DIP fueron sustituidos por jumpers que minimizaban el espacio ocupado, y que a principios de los 90 se empleaban únicamente para seleccionar la frecuencia y el multiplicador de reloj del procesador.

DIP y jumpers

Esta época de transición ha sido más o menos rápida dependiendo del fabricante en cuestión, pero lo cierto es que todos han llegado más tarde o más temprano a una situación de consenso donde la anarquía inicial se ha resuelto en favor de la RAM-CMOS como solución estándar que alberga a todos los parámetros relacionados con la configuración del sistema.

solución estándar

### 2.1.5 Contenidos

Las celdas de memoria de la RAM-CMOS encuentran su correspondencia en un intervalo del espacio total de direcciones del PC, siendo referenciadas desde el procesador como direcciones de 32 bits al igual que cualquier otro área de la memoria del sistema.

direcciones

Puesto que los contenidos de la RAM-CMOS suministran el segmento de datos para cada uno de los programas presentes en la ROM-BIOS, la información que almacena la RAM-CMOS estará emparentada con las tres funciones cubiertas por la BIOS (ver [sección 23.3](#)): (1) albergar las rutinas básicas de entrada/salida, (2) inicializar el PC, y (3) configurar el sistema. Esto nos lleva a diferenciar tres grupos de valores dentro de la RAM-CMOS: Los relacionados con la configuración, que son más dependientes del fabricante, y los que se encuentran ligados a las dos primeras funciones, que conforman información más estandarizada, y por la que vamos a comenzar:

datos

← [pág. 18](#)

tres grupos de valores

- 1 La información estándar está compuesta por instrucciones de control ubicadas en el rango de posiciones de memoria 400h-500h <sup>2</sup>. Este área de memoria se habilitó desde los orígenes del PC para almacenar la información de más bajo nivel del sistema, y conserva su direccionamiento y contenidos originales para garantizar la compatibilidad con los programas más antiguos. La información más relevante que se guarda en este área de memoria es la siguiente:

estándar:

- a) 2 bytes en las direcciones 412h-413h. Contiene el tamaño de la memoria principal del sistema, en Kbytes.
- b) 37 bytes en las direcciones 417h-43Dh. Asociados al teclado. Describen el estado inicial de los LED NUMLOCK y CAPSLOCK que definen la interpretación de las pulsaciones del teclado, la ubicación del buffer del teclado implementado como una cola circular de 32 bytes, y las direcciones que marcan la cabecera y la cola en este buffer.
- c) 30 bytes en las direcciones 449h-466h. Asociados al control de vídeo. Indican cosas como el modo de vídeo activo, la anchura y longitud de la pantalla en los modos de texto y gráficos, el aspecto del cursor y su tamaño, o la selección de la paleta de colores a utilizar durante la visualización.
- d) 5 bytes en las direcciones 46Ch-470h. Contienen la fecha y la hora del sistema.

- memoria

- teclado

- vídeo

- fecha y hora

- 2 La información dependiente del fabricante corresponde a los parámetros de la configuración del sistema. El fabricante de la placa base en la que se incluye la RAM-CMOS habrá considerado una flexibilidad de reconfiguración para el sistema, de tal forma que una mayor complejidad en la versión de BIOS utilizada redundará en una CMOS más extensa.

dependiente del fabricante

Dada la importancia que toda esta información tiene para la integridad del sistema, conforme los sistemas operativos han ido evolucionando se ha venido dotando de mayor grado de protección a todas estas direcciones de memoria. Por ejemplo, el rango de posiciones de memoria en el que se encuentran mapeados los parámetros de la configuración del sistema es irrelevante, pues

protección de acceso

<sup>2</sup>La "h" señala la utilización del código hexadecimal, es decir, el rango comienza en la posición 1024 de memoria principal y finaliza justo 256 bytes después.

los sistemas operativos actuales lo tienen monitorizado para que se pueda acceder a él únicamente a través de los servicios de la BIOS y el RTC. Esto limita el control que el usuario tiene sobre contenidos tan delicados, lo cual no deja de ser una valiosa protección frente a una manipulación perjudicial para nuestro hardware por parte del usuario temerario.

Las BIOS del tipo Plug & Play utilizan, de forma adicional, un segundo área de memoria no volátil donde registran la información de configuración obtenida durante la autoconfiguración de dispositivos. El volumen de información obtenido por este procedimiento se denomina de forma abreviada **ESCD** (*Extended System Configuration Data*), y su tamaño oscila en torno a los 256 bytes. Este área de memoria puede inhibirse desde la opción **RESET CONFIGURATION DATA** del menú **PNP/PCI CONFIGURATION SETUP** de la BIOS (ver [sección 24.3.6](#)).

ESCD

pág. 61

### 2.1.6 Implementaciones

La RAM-CMOS se encuentra integrada dentro de la placa base y viene de serie con ella, no siendo posible sustituirla de forma independiente. A pesar de ello, resulta bastante práctico conocer algunas cosas relativas a su implementación, puesto que de ella dependen ciertas operaciones internas que luego describiremos, como proceder a su borrado cuando se nos olvida la contraseña para entrar en los menús de configuración de la BIOS.

Al contrario que la BIOS, la RAM-CMOS necesita un modesto número de celdas para cumplir con la función que tiene asignada. Dado que la circuitería de la placa base presenta una marcada tendencia a la reducción del espacio ocupado minimizando el número de chips, no es de extrañar que las celdas de la RAM-CMOS hayan buscado cobijo dentro de otro chip más general. Las posibilidades de ubicación han sido tradicionalmente dos:

ubicación

- ❶ La variante más antigua la incluye dentro del chip del reloj en tiempo real (RTC) responsable de la temporización del sistema (ver [sección 23.3.1.2](#)).
- ❷ La alternativa más reciente la sitúa en el interior del puente sur del juego de chips, como consecuencia de la integración a su vez del RTC dentro de éste. Por esta razón, el puente sur y la pila que alimenta a la RAM-CMOS suelen encontrarse muy próximos entre sí en la placa base (ver [foto 23.5](#)).

pág. 22

pág. 17

## 2.2 ▶ La pila

La pila es la responsable de garantizar la alimentación continua y permanente de las celdas de memoria de la RAM-CMOS. Dependiendo de la implementación adoptada para la CMOS, su alimentación autónoma se dispensa también desde un punto diferente de la placa base:

formas:

- ❶ En sus primeros compases de vida como tal dentro de la cuarta generación, la RAM-CMOS se nutría desde una minúscula batería de tamaño ligeramente superior al de su chip (ver [foto 17.13.a](#)).
- ❷ En la mayoría de RAM-CMOS embutidas dentro del reloj en tiempo real en quinta generación, esta pila se encuentra también en el interior de este chip (ver [foto 17.13.c](#)), no siendo posible proceder a su sustitución. Por ello, se contempló para ella una duración superior a los diez años, bastante más de la vigencia que los avances tecnológicos otorgan a los PC.
- ❸ Como variante del esquema anterior, se consideró la inclusión de un acumulador de níquel-cadmio que se recargaba periódicamente con el normal funcionamiento del equipo. El problema aquí era que si el equipo permanecía desconectado por espacio de varios meses, el acumulador se descargaba y los contenidos de la RAM-CMOS se perdían.

p. 29/Vo1.3

- integrada

p. 29/Vo1.3

- acumulador

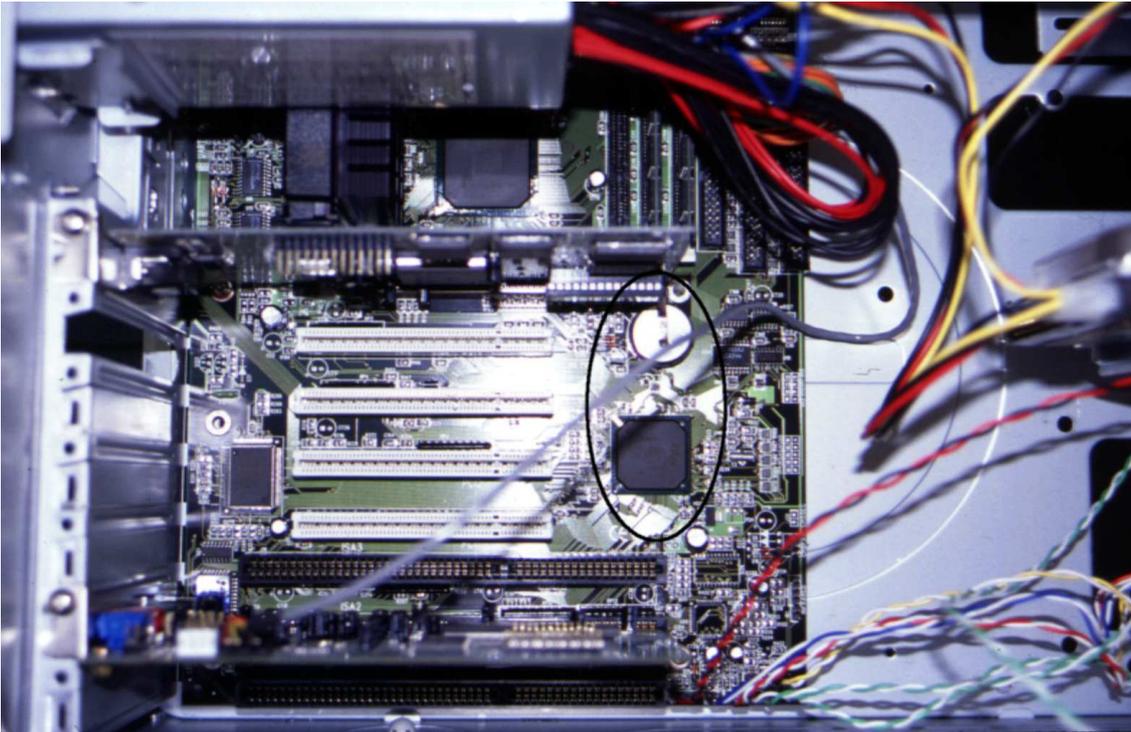


FOTO 23.5: Vista general que se nos ofrece de la pila y la memoria CMOS nada más retirar la carcasa a nuestro equipo. Ambas se encuentran en el centro de la fotografía rodeadas por un óvalo negro, dentro del cual la RAM-CMOS ocupa la parte inferior embutida en el puente sur del juego de chips, y su pila la parte superior. La placa base en cuestión es un modelo para Pentium II con juego de chips 440LX y formato ATX.

- ④ En su forma más contemporánea que contempla la RAM-CMOS como parte del juego de chips, la pila adoptó el formato de una pila de botón similar a la de un reloj de pulsera, pudiendo ser visible y accesible externamente. Su tamaño es similar al de una moneda de un euro (ver fotos 23.5 y 23.6), y su duración estimada se sitúa por encima de los cinco años y aumentando en tiempos más recientes, puesto que las nuevas tecnologías CMOS requieren cada vez un menor consumo.

- de botón

☛ pág. 19  
duración

Además de nutrir a la RAM-CMOS, la pila también proporciona alimentación independiente al reloj de tiempo real (RTC), de donde se emiten los eventos que marcan el paso del tiempo con independencia del estado de encendido o apagado del PC.

Para conocer cómo llevar a cabo la operación de sustitución de la pila y otras labores de mantenimiento asociadas a la misma puede consultarse la [sección 27.1.4](#).

☛ pág. 149

## Borrado de la RAM CMOS

## ◀ 2.3

Esta acción engrosa la lista de cosas que nunca deben efectuarse al manipular el hardware del PC. El único supuesto en que está justificado borrar la CMOS de forma intencionada es el caso en el que hayamos protegido su modificación con una contraseña (normalmente hay una opción para ello en el programa de configuración de la BIOS - ver [sección 24.4.3](#)) y hayamos olvidado esta contraseña.

☛ pág. 71

Un método para borrar los contenidos de la CMOS consiste en retirar durante unos minutos la pila que la alimenta, para lo cual seguiremos las indicaciones dadas en la [sección 27.1.4](#).

☛ pág. 149

manipular  
su jumper  
pág. 19

La mayoría de placas base del año 2000 en adelante permiten realizar la operación de borrado de la CMOS de una forma un poco más elegante: Cambiando de posición el **jumper JCC** situado muy cerca del chip y de su pila (ver parte superior izquierda de la [foto 23.6.b](#)), que se suministra precisamente para cubrir tal función. Tanto en este caso como en el anterior, las operaciones descritas deben llevarse a cabo con la fuente de alimentación del equipo desconectada de su toma de corriente.

descargar  
su capacitor

Algunas CMOS son capaces de retener incluso por espacio de varios días sus contenidos a partir del momento en que se prescinde de la pila o se retire el jumper JCC. Esto es debido a que cuentan con un capacitor cercano de cuya carga se nutren en ausencia de otra alimentación. Si no queremos esperar todo ese tiempo, podemos descargar esta reserva de energía colocando una resistencia de unos 10 K $\Omega$  entre los polos positivo y negativo del zócalo circular con que cuenta la placa base para realizar el contacto con la pila (un detalle de este zócalo vacío puede observarse en la [foto 17.14.b](#)).

p.30/Vol.3

En las placas base más recientes (año 2002 en adelante aprox.), también se ha innovado en este aspecto: Con el jumper JCC en la posición de borrado de la RAM-CMOS, la pulsación del botón de encendido del PC provoca la descarga fulminante de este condensador y el borrado inmediato de datos. El PC no arranca entonces, pero una vez descargado el condensador, devolveremos el jumper a su posición anterior y el sistema volverá a realizar su secuencia de arranque habitual.

confusión

En ocasiones, la memoria CMOS se borra accidentalmente como consecuencia de confundir este jumper JCC con el JAV que existe junto a la Flash BIOS para habilitar su escritura (ver [foto 23.3](#)). Para evitar esta confusión, sólo hay que consultar el manual de la placa base, donde los fabricantes suelen incluir un diagrama con la localización física de cada uno de ellos.

pág. 11

## SECCIÓN 23.3

### Funciones de la BIOS

Con el paso del tiempo, la capa firmware del equipo ha ido creciendo y asentándose. De forma paralela a como hemos visto para la RAM-CMOS, la BIOS ha ido engordando y asumiendo nuevas tareas de mantenimiento y configuración del sistema que antes debían llevarse a cabo directamente sobre el hardware del computador. Esta funcionalidad ha sido implementada de forma decidida por todos los fabricantes, de ahí que, consideraciones de marca aparte, puede decirse que cualquier BIOS actual cubre las **tres funciones** siguientes:

tres funciones:

- rutinas E/S ❶ Proporcionar el sistema básico de rutinas que define la forma de actuar del sistema en respuesta a las peticiones más usuales por parte de los dispositivos de entrada/salida.
- arranque ❷ Controlar la secuencia de arranque e inicialización del sistema.
- configuración ❸ Proporcionar un cómodo interfaz para la fácil selección de parámetros relativos a la configuración del sistema.

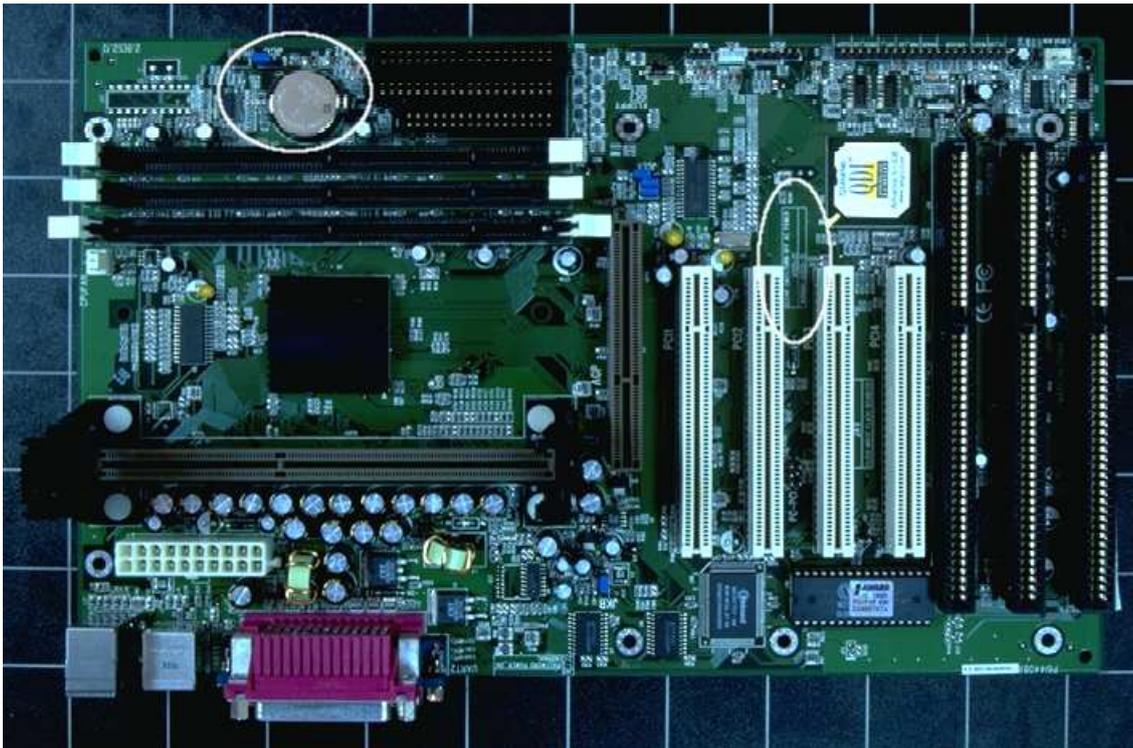
En cualquiera de los tres casos anteriores, lo que la BIOS suministra no es más que un área de memoria física cuyo contenido es un código máquina que debe ser ejecutado por el procesador para cubrir la funcionalidad referida. Cada una de estas funciones será tratada por separado: Las dos primeras, en las [secciones 23.3.1](#) y [23.3.2](#) respectivamente, mientras que a la tercera le vamos a dedicar el [capítulo 24](#) en exclusiva.

pág. 25

pág. 35

#### 3.1 ► Suministrar el sistema básico de entrada/salida

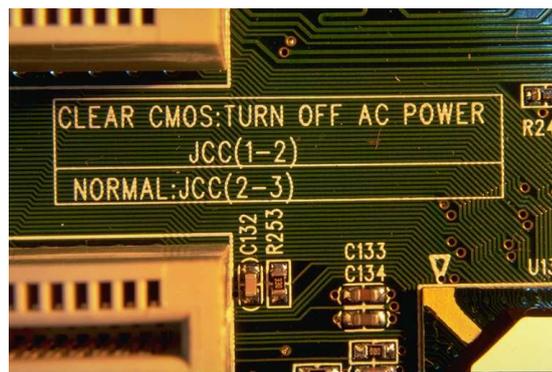
Estamos ante la primera función que asumió la BIOS, y que resulta de extraordinaria importancia para el sistema. Si las interrupciones son el mecanismo que controla todo el funcionamiento



(a)



(b)



(c)

FOTO 23.6: (a) Vista general de una placa base de QDI con juego de chips Apollo Pro 133 de VIA para Pentium III, donde hemos señalado las áreas donde se sitúan la pila de la CMOS junto a su jumper de borrado y la serigrafía que indica el significado de las posiciones para ese jumper. (b) La pila de botón de la CMOS en el centro y el jumper de borrado de sus contenidos (esquina superior izquierda) colocado en las posiciones 2-3 que dejan libre el pin 1 (el que queda más a la derecha). (c) Serigrafía de la placa base que indica que esta posición del jumper obedece a su estado normal (para borrar la CMOS habría que comunicar los pines 1 y 2 dejando libre el tercero).

del PC desde el punto de vista software, la BIOS es la que contiene las rutinas que definen la forma de actuar del computador en respuesta a las interrupciones de más bajo nivel, es decir, las que son más críticas para el funcionamiento del sistema.

función crítica

Las rutinas del sistema básico de entrada/salida se dirigen desde los **vectores de interrupción**, que son 256 punteros o direcciones de memoria de 32 bits ubicados de forma fija en las

vectores de interrupción

posiciones más bajas del mapa de memoria del PC, y que son utilizados tras la generación de la correspondiente interrupción.

Por ejemplo, si se produce la interrupción 7 se dispara la siguiente secuencia de eventos:

- |                        |  |
|------------------------|--|
| salto                  | <ol style="list-style-type: none"> <li>❶ El procesador interrumpe lo que está haciendo y acude a la dirección de memoria 28 (<math>7 \times 4</math> bytes que ocupa cada dirección) para tomar los valores de las direcciones de memoria 28, 29, 30 y 31 con los que compone un puntero que almacena en el registro contador de programa (PC - Program Counter) del interior del procesador.</li> </ol> |
| rutinas de tratamiento | <ol style="list-style-type: none"> <li>❷ El procesador comienza a ejecutar allí la <b>rutina de tratamiento de la interrupción</b> número 7, una secuencia de instrucciones que comienza en la dirección señalizada por el mencionado puntero y que termina con una instrucción especial denominada <b>RTI</b> (retorno de interrupción).</li> </ol>   |
| retorno                | <ol style="list-style-type: none"> <li>❸ La ejecución de esta última instrucción RTI será la encargada de devolver al procesador al estado en que se encontraba antes de que se produjese la interrupción, para que pueda así reanudar el procesamiento de la tarea que se encontraba ejecutando hasta ese momento.</li> </ol>   |

### 3.1.1 Tratamiento de interrupciones hardware y software

**activación** El mecanismo de tratamiento de interrupciones que acabamos de describir puede activarse desde señales de interrupción de muy distinta procedencia, como el botón de encendido del sistema, los dispositivos periféricos (por ejemplo, el teclado o el ratón a su pulsación), o el interior del propio procesador gracias a la ejecución de ciertas instrucciones desde el programa en curso. En función de su origen, conviene diferenciar el tratamiento de las interrupciones hardware y software.

#### ❑ Interrupciones hardware

**PIC 8259** Las interrupciones hardware son generadas desde patillas físicas directamente conectadas al controlador de interrupciones. En los primeros procesadores, como el 8088 y el 8086, este controlador se implementaba mediante un chip separado, el **PIC 8259** del Intel, que recepcionaba ocho fuentes de interrupción diferentes y permitía programar sus prioridades y enmascaramientos. Posteriormente, con la llegada del 80286, su funcionalidad se amplió para dar cobertura a 15 fuentes de interrupción hardware, para lo cual se montaron dos chips PIC 8259 conectados en cascada mediante una configuración de maestro y esclavo (consultar [figuras 24.2 y 24.3](#)).

[pág. 63](#) ➔

[pág. 64](#) ➔

**ubicación del PIC**

Los sistemas contemporáneos a éste que alcanzan ya nuestros días mantienen esa misma funcionalidad, aunque el PIC 8259 ya no existe como tal, sino que se encuentra formando parte de la circuitería del puente norte del juego de chips de la placa base.

**conflictos**

Esta funcionalidad lleva tiempo pidiendo a gritos una ampliación, ya que las 15 líneas de interrupción existentes en la actualidad son claramente insuficientes a la vista del creciente número de periféricos que pretenden conectarse al sistema. De hecho, la asignación de estas líneas, denominadas IRQ (consultar [sección 24.3.6](#)), es uno de los aspectos que más conflictos produce en la actualidad a la hora de configurar el sistema, ya que para colmo su asignación a dispositivos dista mucho de ser libre.

**IRQ #**

[pág. 61](#) ➔

**soluciones**

La llegada de los sistemas operativos PnP ha simplificado el proceso de asignación, aunque no ha garantizado la ausencia de conflictos. Algunas placas base sí han aportado algo más, incorporando un nivel firmware adicional que presenta multitud de líneas IRQ virtuales a la capa software, para luego encargarse de mapearlas de forma transparente sobre las líneas físicas existentes.

### □ Interrupciones software

Si el mecanismo de vectores de interrupción que utilizamos para implementar las interrupciones consideró 256 interrupciones y el controlador de interrupciones que acabamos de describir contempla únicamente la generación de 15 interrupciones, cabe preguntarse cuáles son los mecanismos encargados de generar las 241 interrupciones restantes.

La respuesta la encontramos en las interrupciones software, producidas todas ellas por la ejecución de una instrucción especial del procesador, cuya sintaxis más extendida en lenguaje ensamblador utiliza el mnemotécnico **INT #** seguido del número de interrupción como único argumento.

Si habíamos asociado las interrupciones a eventos inesperados y asíncronos del sistema, puede sorprendernos el hecho de que haya interrupciones que puedan ser activadas premeditadamente por la normal ejecución de un programa. Pero si seleccionamos al azar una interrupción que ocurra en el PC, lo realmente extraño será que resulte ser de tipo hardware, porque las interrupciones software son tan habituales que constituyen la forma de trabajo más habitual para nuestro procesador. De hecho, los servicios de la BIOS pueden verse como la capa más baja del sistema operativo, encima de la cual se ha construido durante veinte años todo el software existente en la actualidad.

La razón por la que los servicios de la BIOS fueron implementados en forma de rutinas de tratamiento para alguna interrupción hay que buscarla en la extraordinaria analogía sintáctica y semántica de la instrucción **INT #** con la instrucción **CALL proced**<sup>3</sup>, un mecanismo con el que se encuentran altamente familiarizados los programadores de los lenguajes de alto nivel para insertar llamadas a procedimientos y subrutinas en el código de las aplicaciones.

No obstante, en **INT #** confluyen tres matices diferenciadores que conviene tener suficientemente claros:

- ❶ Al encontrarnos más cerca del nivel hardware de la máquina, muchas de las rutinas contienen instrucciones dependientes de los dispositivos con los que interactúan. Las rutinas que exhiban esta dependencia no son proporcionadas por la BIOS, sino desde el correspondiente *controlador de dispositivo* ó *driver* del sistema operativo.
- ❷ No tenemos que preocuparnos de escribir nosotros el código máquina de la rutina correspondiente, puesto que en cualquier caso ésta se encuentra grabada ya en la BIOS como parte de su implementación. En el caso de que la rutina sea suministrada o actualizada por el sistema operativo como parte de algún controlador de dispositivo, será también el propio sistema operativo el encargado de actualizar los vectores de interrupción correspondientes para que apunten a las áreas de memoria donde haya ubicado dichas rutinas.
- ❸ El paso de parámetros a estas rutinas se establece desde los registros de propósito general del procesador. Cada rutina presenta un interfaz propio preestablecido en función de la información que debe proporcionarle el programador como entrada y los datos que la rutina debe devolverle como salida. Eso sí, el primer registro de propósito general de 32 bits, denominado **AX**, tiene en este caso una misión específica: su mitad más significativa (también denominada **AH**) selecciona el número de servicio que se quiere utilizar dentro de esa interrupción.

Los servicios asociados a una misma interrupción no son más que una forma de agrupar tareas similares en torno a un mismo número de interrupción, y sirven para estructurar un poco el conjunto de funciones cubiertas por la BIOS al tiempo que se consigue extender el límite de 256 rutinas que impuso el mecanismo de vectores de interrupción originalmente implementado sobre posiciones fijas de memoria.

<sup>3</sup>Hemos utilizado la sintaxis del lenguaje FORTRAN por ser el que más se utilizaba cuando aparecieron las rutinas de la BIOS, pero lo cierto es que la semántica de esta instrucción sigue vigente en cualquier lenguaje de alto nivel contemporáneo.

INT #

proliferación

implementación

tres matices

- dependencia  
(en rutinas)- ubicación  
(de rutinas)- interfaz  
(con rutinas)

servicios

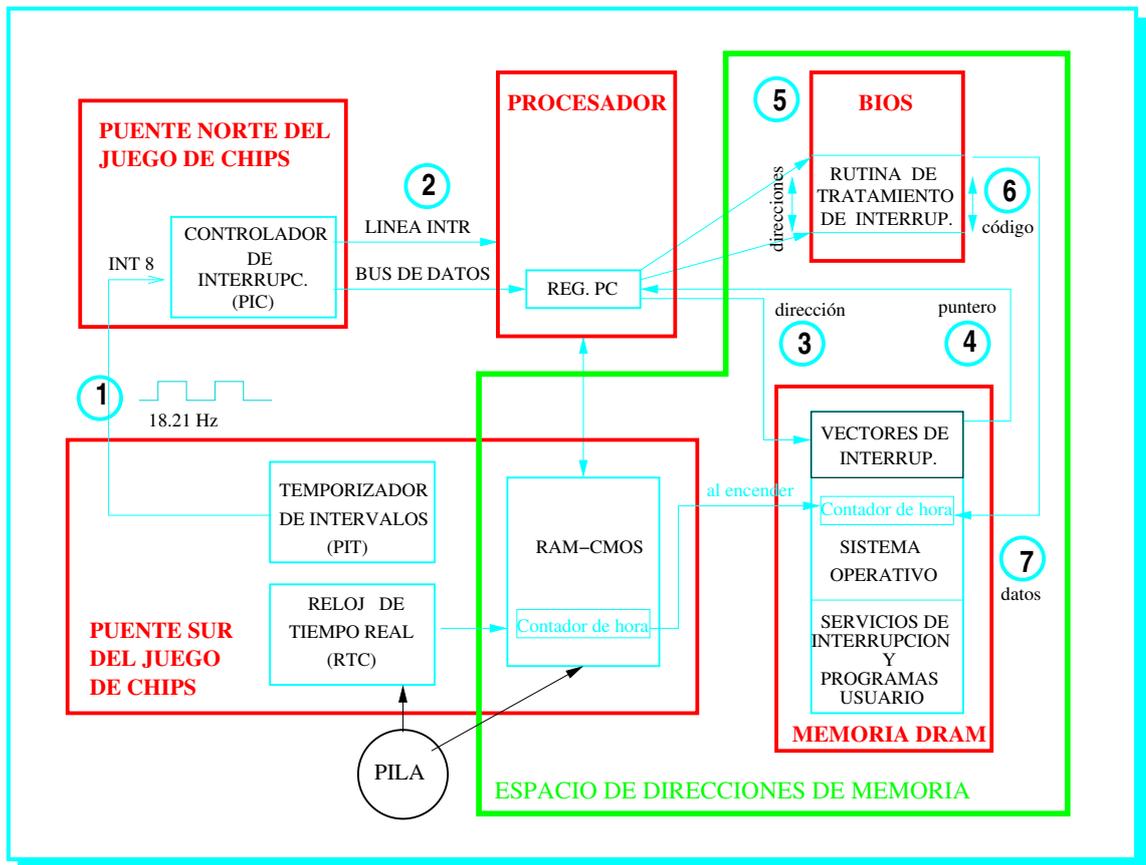


FIGURA 23.1: Círcuitería implicada en la continua actualización de la fecha y hora del sistema mediante una serie de servicios de interrupción y un contador alojado en la RAM-CMOS.

### 3.1.2 Ejemplo: Actualización del reloj del sistema

Para facilitar la comprensión de las distintas variantes de interrupción y la forma en que los contenidos de la ROM-BIOS y la RAM-CMOS son utilizados por éstas, vamos a describir el funcionamiento de la rutina asociada a la interrupción hardware número 8, que se encuentra mapeada sobre la línea IRQ 0 del controlador de interrupciones (PIC - ver [figura 24.2](#)), y encargada de registrar y mantener la hora y la fecha del sistema en la posición de memoria 46Ch cuando se encuentra encendido mediante el controlador de intervalos programable descrito en la [sección 17.2.3](#).

Con posterioridad, describiremos el conjunto de seis servicios asociados a la interrupción software número 26. Estos servicios se encargan de hacer visibles los valores registrados en 46Ch a la capa software del sistema. Dado que estos servicios son completamente independientes de cualquier dispositivo hardware, sus rutinas asociadas no forman parte del controlador de ningún dispositivo, sino que constituyen uno de los ejemplos más genuinos de información contenida en la BIOS del sistema.

#### ❑ Interrupción 8: Registrando internamente el paso del tiempo.

Nos encontramos frente a una interrupción hardware en la que se encuentran involucrados cinco componentes hardware, aportando cada uno de ellos la siguiente funcionalidad:

- rutina      ❶ La BIOS. Contiene la rutina de tratamiento para la interrupción hardware 8, utilizando para

rutina  
de int. HW  
pág. 63

p. 12/Vo1.3

servicios  
de int. SW

ello un grupo de posiciones de memoria virtuales consecutivas. Supondremos que la primera de estas posiciones es la 01234567h (esta posición es variable en cada PC dependiendo de la BIOS de que se disponga).

- 2 El puente sur del juego de chips.** Emite pulsos para la generación de una interrupción con una cadencia de 18.21 Hz, ayudándose de la intervención del temporizador de intervalos programable (PIT - ver [sección 17.2.3](#)). La [figura 17.1](#) sintetiza su contribución desde la generación de la señal de reloj en el oscilador de cuarzo y el chip generador de reloj (circuito PLL - ver [foto 17.3](#)) que proporciona la frecuencia de funcionamiento a la placa base.
- 3 El controlador de interrupciones.** Incluido en el puente norte del juego de chips, recoge la señal procedente del PIT por su patilla IRQ 0, que corresponde a la interrupción hardware número 8, y resuelve las prioridades y enmascaramientos que le han sido programados para esa interrupción. Seguidamente, informa al procesador de que se ha producido una interrupción, activando para ello su línea INTR y volcando al bus de datos el número de interrupción.
- 4 El procesador** percibe la activación de su patilla INTR, acudiendo al bus de datos para recepcionar el número de la interrupción que se acaba de producir. Una vez hecho esto, comienza el tratamiento de la interrupción, que consta de la siguiente secuencia de pasos:
- El procesador salva el estado de sus registros internos en un área de memoria reservada para tal efecto y que recibe el nombre de segmento de pila <sup>4</sup>. Esto permitirá al procesador reanudar la ejecución del programa con el que se encontraba trabajando antes de que ocurriera la interrupción.
  - A partir del número de interrupción recogido del bus de datos (el 8 en este caso), se direcciona a memoria a la posición absoluta  $8 \times 4 = 32$ , que es donde se encuentra el vector de interrupción para esa interrupción. Del contenido del byte 32 y los tres siguientes, el procesador compone el correspondiente puntero, que corresponde en este caso a la dirección 01234567Ch en nuestro caso.
  - Esta dirección se carga en el registro contador de programa (PC) del procesador, lo que funcionalmente se traduce en el comienzo de la ejecución de la rutina de tratamiento para la interrupción 8 ubicada en esa dirección. La tarea que lleva a cabo esta rutina es tan sencilla como incrementar el valor de un contador que registra el número de pulsos generados por el PIT, y a partir del cual puede obtenerse la hora en cada instante con una resolución de la decimo octava parte del segundo (18.21 Hz ó 55 milisegundos). Cuando el PC se encuentra apagado, esta nimia tarea se lleva a cabo desde el controlador de tiempo real (RTC) con una resolución superior (1 KHz - ver [sección 17.2.4](#)) y sin necesidad de salir del puente sur del juego de chips, puesto que allí también se encuentran físicamente las celdas de la RAM-CMOS donde se registra el reloj interno del sistema. El valor de este reloj es el que se copia a la dirección 46h de memoria principal cada vez que se enciende el sistema. Mientras está apagado, el RTC se alimenta de la energía que suministra la pila de litio de la placa base.
  - La rutina termina ejecutando la instrucción RTI (retorno de tratamiento de interrupción), que recupera desde la pila el contenido de los registros internos para proseguir con la ejecución del programa que se encontraba activo antes de que se generase la interrupción. De esta manera, el programa se reanuda en el mismo punto en que fue interrumpido, y con los mismos valores que contenían los registros en aquel momento.
- 5 La RAM-CMOS** reserva un entero de 4 bytes donde almacenar el contador incrementado por la rutina de la BIOS que se ejecutó como tratamiento de la interrupción 8.

<sup>4</sup>La denominación de pila hace referencia a la forma en la que se manejan los datos en esta zona de memoria. Desde una perspectiva software, una pila representa una estructura de datos en la que la información únicamente puede ser extraída en el orden inverso en el que ha sido introducida.

pulsos

← p. 12/Vol.3

← p. 11/Vol.3

← p. 10/Vol.3

notificación

tratamiento de  
la  
interrupción:  
salvar

direccionar

ejecutar

← p. 13/Vol.3

recuperar

La secuencia completa de pasos que se produce para contabilizar en este entero de la RAM-CMOS el número de pulsos procedentes del PIT se muestra en la [figura 23.1](#).

#### □ Interrupción 26: Lectura y escritura de la hora y la fecha del sistema

instancias

A diferencia de la interrupción número 8, que se genera por una patilla hardware, la interrupción número 26 es de tipo software, o sea, se produce por la ejecución de la instrucción **INT 26** desde el programa en curso. Esto permite instanciar distintos servicios de interrupción (que se encuentran numerados desde el 0 al 5 en este caso) sin más que cargar su número en el registro AH con antelación a la ejecución de INT 26.

#### Servicios 0 y 1: Lectura y escritura de la hora en formato contador

El contador que registra en la RAM-CMOS el número de pulsos producidos por el reloj de tiempo real puede ser leído o modificado por el procesador a petición de los servicios 0 y 1, respectivamente. Su valor pasará al registro CX tras finalizar el servicio 0 o se colocará en éste antes de solicitar el servicio 1, mientras que el registro AL indicará con un valor distinto de cero si se han sobrepasado las 24 horas, en cuyo caso hay que proceder a actualizar la fecha del sistema.

#### Servicios 2 y 3: Lectura y escritura de la hora en formato hh:mm:ss

lectura

Los servicios 2 y 3 son muy similares a los dos anteriores, sólo que en lugar de trabajar con el valor del contador, lo hacen con los valores relativos a la hora, los minutos y los segundos, más sencillos de manejar si es el usuario el que hace uso de ellos. Así, el servicio 2 proporciona la hora, los minutos y los segundos en los registros CH, CL y DH, respectivamente, obteniendo dichos valores sin más que efectuar las siguientes operaciones sobre el contador anterior:

```
Hora          = contador / 65543.33
SobranterHor = contador % 65543.33
Minutos       = SobranterHor / 1092.38
SobranterMin  = SobranterHor % 1092.38
Segundos      = SobranterMin / 18.21
SobranterSeg  = SobranterMin % 18.21
Centesimas    = Parte_Entera(SobranterSeg * 100)
```

donde / y % son los operadores de división entera y resto de la división entera, respectivamente. Los valores 1092.38 y 65543.33 salen de interpretar la frecuencia de actualización del contador y las típicas equivalencias horarias. Es decir, si el contador se actualiza 18.21 veces en un segundo, lo hará  $18.21 * 60 = 1092.38$  veces en un minuto, y  $1092.38 * 60 = 65543.33$  veces en una hora <sup>5</sup>.

escritura

Por su parte, el servicio 3 permite escribir el valor del contador dando también los valores relativos a la hora, minutos y segundos. En este caso, la rutina de la BIOS asociada a este servicio deberá realizar las siguientes operaciones:

```
contador = (Hora * 65543.33) + (Minutos * 1092.38)
+ (Segundos * 18.21) + (Centesimas * 0.182)
```

aunque podemos apreciar que la rutina no permite ajustar el valor del contador con una resolución de centésimas de segundo.

#### Servicios 4 y 5: Lectura y escritura de la fecha del sistema.

En base a la frecuencia de actualización de 18.21 veces por segundo, el contador anterior alcanza

<sup>5</sup>Los valores no son exactos por los redondeos efectuados.

la medianoche para un valor de 1573040, evento que se encuentra registrado de forma permanente en otra posición de memoria del sistema, la 470h, y que además provoca la actualización de la fecha del sistema en otras posiciones de la RAM-CMOS.

Cuando el sistema operativo necesita conocer la fecha del sistema para registrarla en alguna de sus operaciones (o simplemente mostrarla al usuario), hace uso del servicio 4, que lee esta fecha y la sitúa en los registros CX y DX de acuerdo a la siguiente correspondencia: DL para el día, DH para el mes, CH para las dos primeras cifras del año y CL para las dos últimas cifras del año (a pesar de su avanzada edad, las rutinas de la BIOS fueron de los pocos entes software que se diseñaron libres del efecto 2000).

De forma similar, cuando queramos cambiar la fecha del sistema, podemos hacer uso del servicio 5, donde, con antelación a la ejecución de la instrucción INT 26, introduciremos un 5 en el registro AH y la nueva fecha en los registros CX y DX siguiendo la misma correspondencia que para el servicio 4 anteriormente descrito.

Existen algunos servicios horarios un poco más especializados, pero su descripción es muy similar a los seis que hemos incluido aquí de forma ilustrativa.

## Iniciación del sistema

## ◀ 3.2

Si no prestamos atención a la secuencia de eventos que se suceden en el sistema desde que pulsamos su botón de encendido, podemos tener la falsa impresión de que el equipo se encuentra en todo momento bajo el control del sistema operativo.

Pero si nos ponemos a pensar con un poco más de detenimiento, advertiremos que el sistema operativo es un programa, y como tal, debe ser cargado en memoria desde el dispositivo que lo contiene (disco duro, CD-ROM, ...) antes de quedar dispuesto para asumir definitivamente el control de nuestro PC.

Esto significa que existe un período de tiempo inicial en el que el sistema debe autogobernarse para evolucionar hacia un estado de madurez suficiente como para delegar la responsabilidad de su gestión a la capa software. Este período de ajuste se denomina **iniciación del sistema**, y la lista de eventos que se suceden durante el mismo se conoce como **secuencia de arranque**.

iniciación  
del sistema

En la transferencia de responsabilidades desde el hardware al software que tiene lugar durante la iniciación del sistema, la capa firmware actúa como lógica intermediaria según se desprende de su posición natural, y aquí es donde la BIOS desempeña la primera de sus funciones: Albergar el código máquina del programa que establecerá la secuencia de arranque y asumirá el control temporal del equipo durante su iniciación.

La secuencia de arranque apenas dura unos breves instantes, y consta de la carga en memoria y posterior ejecución de dos códigos, por este orden: El programa inicial y el programa de arranque del sistema operativo. Aunque probablemente estemos aburridos de presenciarla in situ, merece la pena describirla para comentar algunos de sus aspectos más interesantes. Tras una pulsación del botón de encendido del equipo por parte del usuario, ésta es la secuencia ordenada de eventos que le sucede de forma inmediata:

secuencia  
de arranque

- ❶ **Inicialización de los registros internos del procesador.** Una serie de señales eléctricas viajan a través del juego de chips de la placa base camino del procesador. Estas señales se encuentran lideradas por RESET, que presenta un significado muy especial, pues su posterior transición a la baja llevará consigo la puesta a cero del conjunto de información que el procesador alberga en su interior, entre la que cabe reseñar:

puesta a cero

- a) Sus bancos de registros de propósito general para computación entera, multimedia y de punto flotante.

- b) Sus cachés de primer y segundo nivel integradas, tanto para datos como para instrucciones.
- c) Sus mecanismos de predicción y traducción efectivas: BTB y TLB, respectivamente.
- d) Los registros específicos para cada modelo comercial según sus rasgos particulares.



### Ejemplo 23.2: LOS REGISTROS MSR DE INTEL

En los procesadores de Intel, los registros específicos a que nos referíamos en el párrafo anterior se denominan **MSR (Model Specific Registers)**, y son en su mayoría contadores que recolectan por separado una serie de eventos significativos del sistema (fallos a caché, fallos a TLB, fallos de página, fallos de predicción de salto, etc). Entre todos ellos destacaríamos el registro EBL\_CR-POWERON, en cuyos bits 22 al 24 se almacena el multiplicador que programa el circuito PLL interno del procesador para determinar la que será su frecuencia de trabajo hasta que RESET vuelva a activarse. Este es el único instante en que dichos bits admiten una escritura, y el valor que se grabe en ellos dependerá del estado de una serie de pines en ese preciso instante: LINT, BF o FID según el modelo concreto de procesador a que nos refiramos y su fabricante, tal y como detallamos para los modelos de Intel y AMD en la [sección 30.4.3.3](#).

- ② **Carga del programa inicial.** A continuación, la circuitería generará directamente la dirección física asociada a la posición de la BIOS donde comienzan las instrucciones responsables de cargar en memoria el programa inicial, o **IPL (Initial Program Load)**. Esta dirección es la 000FFFF0h, y se mantiene fija desde el nacimiento del PC por compatibilidad con los sistemas más antiguos.

IPL

direcciones  
físicas y  
virtuales

Nos encontramos ante una de las escasas situaciones en las que el procesador trabaja directamente con direcciones físicas. Las áreas de memoria del sistema operativo y del espacio de usuario son referenciadas en cualquier PC como direcciones virtuales para facilitar una eventual reubicación del código y permitir la ejecución de programas más grandes que el espacio físico libre en memoria, realizándose la traducción virtual a física mediante una tabla de páginas que es propiedad del sistema operativo.

Como el programa inicial se ejecuta por debajo del sistema operativo y antes de que éste se encuentre activo, debe prescindir de todas las facilidades que éste proporciona al resto de programas en general. Este matiz diferenciador es el que conduce a que las posiciones de memoria de la BIOS se desmarquen del resto en su implementación, teniendo siempre una dirección preestablecida que permita al hardware acceder a ella de forma directa.

- ③ **Ejecución del programa inicial.** El procesador comienza a ejecutar el código del programa inicial a partir de las direcciones donde se encuentra alojada la BIOS de forma permanente, esto es, en la parte más alta del primer megabyte de memoria direccionable (rango 896K-1024K para la BIOS del sistema y otros rangos inferiores para las BIOS de los dispositivos - consultar el mapa de memoria del PC que adjuntamos en la [figura 16.5](#)).

p. 227/Vo1.2

Dado que tanto la ubicación de la BIOS como su tamaño quisieron contemplarse con carácter variable en aras a garantizar la compatibilidad con subsiguientes sistemas de mayor complejidad, el único requisito que se exige a la BIOS aquí es que coloque como primera posición física del programa inicial una instrucción de salto ("jump") para que el programa prosiga la ejecución en la dirección concreta donde el modelo de BIOS con que venga equipado nuestro PC tenga ubicado su programa de arranque.

Pitidos breves	Cód. POST	Operación realizada por las rutinas de autotesteo	Pitidos breves	Cód. POST	Operación realizada por las rutinas de autotesteo
1-1-1-3	02h	Verificar que nos encontramos en modo real	2-4-1-3	72h	Comprobar errores de configuración
1-1-2-1	04h	Obtener el tipo de CPU presente en el sistema	2-4-2-1	74h	Comprobar el reloj de tiempo real (RTC)
1-1-2-3	06h	Inicializar el hardware del sistema	2-4-2-3	76h	Comprobar errores procedentes del teclado
1-1-3-1	08h	Inicializar los registros internos del juego de chips	2-4-4-1	7Ch	Activar los vectores de interrupción hardware
1-1-3-2	09h	Activar el flag POST	2-4-4-3	7Eh	Testear el coprocesador matemático (si existe)
1-1-3-3	0Ah	Inicializar los registros de la CPU	3-1-1-1	80h	Desactivar los puertos E/S internos
1-1-4-1	0Ch	Inicializar las cachés	3-1-1-3	82h	Detectar e instalar los puertos RS232 externos
1-1-4-3	0Eh	Inicializar el sistema de E/S	3-1-2-1	84h	Detectar e instalar los puertos paralelo externos
1-2-1-1	10h	Inicializar la gestión de consumo	3-1-2-3	86h	Reinicializar los puertos de E/S internos
1-2-1-2	11h	Cargar registros alternativos con valores iniciales	3-1-3-1	88h	Inicializar el área de datos de la BIOS
1-2-1-3	12h	Saltar a UserPatch0	3-1-3-3	8Ah	Inicializar el área de datos extendida de la BIOS
1-2-2-1	14h	Inicializar el controlador de teclado	3-1-4-2	8Ch	Inicializar el controlador de la disquetera
1-2-2-3	16h	Verificar suma de comprobación en chip BIOS	3-2-1-1	90h	Inicializar el controlador de disco duro
1-2-3-1	18h	Inicialización del temporizador PIT 8254	3-2-1-2	91h	Inicializar el bus del controlador de disco duro
1-2-3-3	1Ah	Inicialización del controlador DMA 8237	3-2-1-3	92h	Saltar a UserPatch2
1-2-4-1	1Ch	Inicialización del controlador PIC de interrupciones	3-2-2-1	94h	Desactivar la línea de dirección A20
1-3-1-1	20h	Testear el refresco de memoria principal	3-2-2-3	96h	Borrar el registro de segmento ES
1-3-1-3	22h	Testear el controlador de teclado 8742	3-2-3-1	98h	Buscar las BIOS opcionales
1-3-2-1	24h	Activar el registro de segmento ES a 4 Gbytes	3-2-3-3	9Ah	Ensombrear (copiar a memoria) BIOS opcionales
1-3-3-1	28h	Detectar automáticamente el tamaño de memoria	3-2-4-1	9Ch	Activar la gestión de consumo
1-3-3-3	2Ah	Borrar los primeros 512 Kbytes de memoria principal	3-2-4-3	9Eh	Habilitar las interrupciones hardware
1-3-4-1	2Ch	Testear líneas de direcciones hasta 512 Kbytes	3-3-1-1	A0h	Activar la hora del día
1-3-4-3	2Eh	Testear datos de los primeros 512 Kbytes	3-3-1-3	A2h	Comprobar el bloqueo del teclado numérico
1-4-1-3	32h	Testear la frecuencia de reloj del bus local	3-3-3-1	A8h	Borrar el aviso de "Press F2 to enter Setup"
1-4-2-4	37h	Reinicializar el juego de chips de la placa base	3-3-3-3	AAh	Comprobar si se pulsó F2 para entrar en Setup
1-4-3-1	38h	Ensombrear (copiar a memoria) la BIOS del sistema	3-3-4-1	ACH	Entrar en el Setup de la BIOS
1-4-3-2	39h	Reinicializar las cachés	3-3-4-3	AEh	Borrar el flag que señala estar dentro de POST
1-4-3-3	3Ah	Detectar automáticamente el tamaño de caché	3-4-1-1	BOh	Comprobar errores
1-4-4-1	3Ch	Configurar los registros avanzados del juego de chips	3-4-1-3	B2h	Fin de POST; prepararse para arrancar el S.O.
1-4-4-2	3Dh	Cargar los regs. alternativos con valores de RAM-CMOS	3-4-2-1	B4h	Emitir un breve pitido por el altavoz
2-1-1-1	40h	Activar la velocidad inicial de la CPU	3-4-2-3	B6h	Comprobar la contraseña de arranque (opcional)
2-1-1-3	42h	Inicializar los vectores de interrupción	3-4-3-1	B8h	Borrar la tabla de descriptores globales
2-1-2-1	44h	Inicializar las interrupciones de la BIOS	3-4-4-1	BCh	Borrar los comprobadores de paridad
2-1-2-3	46h	Comprobar el aviso de copyright de la BIOS	3-4-4-3	BEh	Borrar la pantalla (opcional)
2-1-2-4	47h	Inicializar el gestor para las BIOS opcionales PCI	3-4-4-4	BFh	Comprobar virus y recordatorios de seguridad
2-1-3-1	48h	Contrastar la configuración de vídeo con RAM-CMOS	4-1-1-1	C0h	Intentar arrancar con INT 19h
2-1-3-2	49h	Inicializar el bus PCI y sus dispositivos	4-2-1-1	D0h	Error en el manejo de interrupciones
2-1-3-3	4Ah	Inicializar los adaptadores de vídeo que estén presentes	4-2-1-3	D2h	Error de interrupción desconocida
2-1-4-1	4Ch	Ensombrear (copiar a memoria DRAM) la BIOS de vídeo	4-2-2-1	D4h	Error de interrupción pendiente
2-1-4-3	4Eh	Mostrar en pantalla el aviso de copyright	4-2-2-3	D6h	Error en la inicialización de las BIOS opcionales
2-2-1-1	50h	Mostrar en pantalla el tipo y velocidad de la CPU	4-2-3-1	D8h	Error de reinicialización
2-2-1-3	52h	Testear el teclado	4-2-3-3	DAh	Mover el bloque extendido
2-2-2-1	54h	Activar la pulsación de teclas si el teclado responde OK	4-2-4-1	DCh	Error en reinicialización de E/S
2-2-2-3	56h	Activar el teclado	4-3-1-3	E2h	Inicializar el juego de chips de la placa base
2-2-3-1	58h	Testear las interrupciones inesperadas	4-3-1-4	E3h	Inicializar el contador de refresco de memoria
2-2-3-3	5Ah	Mostrar el mensaje "Press F2 to enter SETUP"	4-3-2-1	E4h	Verificar la Flash-BIOS
2-2-4-1	5Ch	Testear la DRAM en el rango 512-640 Kbytes	4-3-2-2	E5h	Verificar el estatus hardware de la Flash-BIOS
2-3-1-1	60h	Testear la memoria expandida	4-3-2-3	E6h	La Flash-BIOS ha sido validada correctamente
2-3-1-3	62h	Testear líneas de direcciones para memoria extendida	4-3-2-4	E7h	Testear por completo la memoria principal
2-3-2-1	64h	Saltar a UserPatch1	4-3-3-1	E8h	Realizar la inicialización de OEM
2-3-2-3	66h	Configurar los registros avanzados de caché	4-3-3-2	E9h	Inicializar el controlador de interrupciones
2-3-3-1	68h	Habilitar la jerarquía de cachés (integradas y externas)	4-3-3-3	EAh	Leer el código bootstrap
2-3-3-3	6Ah	Mostrar en pantalla el tamaño de las cachés externas	4-3-3-4	EBh	Inicializar todos los vectores
2-3-4-1	6Ch	Mostrar mensaje de áreas Shadow	4-3-4-1	ECh	Arrancar el programa Flash
2-3-4-3	6Eh	Mostrar segmentos no desechables	4-3-4-2	EDh	Inicializar el dispositivo de arranque del S.O.
2-4-1-1	70h	Mostrar en pantalla mensajes de error	4-3-4-3	EEh	El código de arranque fue leído correctamente

TABLA 23.1: Códigos POST escritos en el puerto 80 durante el autotesteo de dispositivos y sus pitidos asociados en caso de anomalía hardware para la versión 4.0 de las BIOS de Phoenix.

⚙️ **Autotesteo de dispositivos.** La primera parte del programa inicial realiza el autotesteo de los dispositivos más básicos, operación que de forma abreviada recibe el nombre de **POST (Power-On Self Test)**.

POST

Se trata de un compendio de rutinas que se encargan de comprobar el buen estado de salud de la circuitería del sistema, y que realizan de forma secuencial la lista de operaciones que adjuntamos en la [tabla 23.1](#) (como podemos apreciar, POST se extiende más allá de este paso, hasta completarse la secuencia de arranque del PC). A cada nueva operación, POST escribe en el puerto 80 correspondiente al rango de direcciones de entrada/salida el código de un byte mostrado en dicha tabla, con lo que monitorizando el valor de dicho puerto podemos disponer de una exhaustiva traza de la secuencia de arranque (de hecho, existen tarjetas ISA y PCI que presentan el valor de este puerto en un display de dos dígitos hexadecimales para poder resolver mejor posibles anomalías en nuestro sistema).

POST dura hasta el paso 10

En caso de encontrar alguna anomalía, el sistema informará del error, pero como en la mayoría de los posibles eventos la tarjeta gráfica aún no ha sido inicializada y las funciones de visualización en el monitor no se encuentran disponibles, el código de error se emite por el altavoz interno que suele incluir la carcasa del equipo. Este código POST se codifica como una secuencia de pitidos de distinta duración y frecuencia de repetición que depende del

código de error

fabricante de la BIOS, tal y como hemos recogido en la [tabla 23.1](#) para el fabricante Phoenix y en la [tabla 28.1](#) para los otros dos grandes fabricantes de BIOS, AMI y Award.

Si queremos agilizar el arranque del sistema, la propia BIOS nos brinda la posibilidad de simplificar las comprobaciones que se realizan en este paso (se prescinde de las más superfluas y tediosas, como el testeado de las celdas de memoria principal), para lo cual deberemos activar la opción **QUICK POWER ON SELF TEST** dentro del menú **BIOS FEATURES SETUP** de configuración del sistema (ver [sección 24.3.3](#)).

- 5 **Búsqueda de BIOS adicionales.** El siguiente paso del programa inicial de la BIOS consiste en rastrear las posiciones de memoria en las que tradicionalmente se encuentran ubicadas las BIOS específicas de los distintos dispositivos. Lo primero que encuentra es siempre la BIOS de la tarjeta gráfica, ubicada en el intervalo comprendido entre las posiciones 000C0000h y 000C7800h, ejecutando el programa inicial de esta BIOS, lo que operativamente corresponde a una inicialización de la tarjeta gráfica. Estas instrucciones son las primeras que pueden escribir en pantalla, con lo que, de emitir algún mensaje informativo acerca de las prestaciones gráficas del equipo, será lo primero que veremos en pantalla.

A continuación, el sistema prosigue la búsqueda de otras BIOS en el rango comprendido entre las posiciones 000C8000h y 000DF800h, a incrementos de 2 Kbytes. La primera que encuentra siempre es la del disco duro IDE/ATA (pues se sitúa sobre la posición inicial 000C8000h), procediendo a ejecutarla a continuación para inicializar este dispositivo. De encontrar alguna BIOS adicional, prosigue de forma similar ejecutando su respectivo programa inicial. El [ejemplo 24.1](#) indica las posiciones de éstas.

Puede ser que el periférico, por su constitución, no disponga de BIOS propia, pero que requiera una mínima inicialización para quedar operativo entretanto se carga el sistema operativo. Así ocurre con el teclado, al que la BIOS presta un *driver* básico para que podamos utilizarlo antes de que quede definitivamente en manos del sistema operativo. La [sección 16.3.1.5](#) amplía este proceder desde la perspectiva de un teclado USB.

- 6 **Visualización de la pantalla inicial.** El programa inicial visualiza a continuación la pantalla de presentación, con lo que comenzamos a ver los primeros síntomas de vida de nuestro equipo: Aparecerán en nuestro monitor unos caracteres indicando, en la parte superior, el fabricante y la versión de la BIOS de que disponemos, así como el logotipo del fabricante del PC y/o la placa base, mientras que en la parte inferior se muestra la combinación de teclas que hay que pulsar para entrar en el menú de configuración del sistema ó *Setup* (ver [tabla 24.1](#)) y el número de serie del chip BIOS.

- 7 **Emisión del informe sobre el estado de los dispositivos.** El sistema prosigue realizando más testeos, como el de las celdas de memoria principal, y concluye emitiendo un breve informe del estado de los dispositivos. Si la BIOS detecta alguna anomalía, emite un breve mensaje de error y seguidamente ejecuta una instrucción HALT que detiene la actividad del procesador. En caso contrario, la BIOS realiza un inventario algo más exhaustivo de los recursos de que dispone el sistema, informando al usuario de las prestaciones que quedan a su disposición.

Esta información se presenta en una primitiva tabla de caracteres ASCII que incluye información como el tipo de CPU, su frecuencia de reloj, las unidades de disco reconocidas (flexibles, duros y CD-ROM), el tamaño de caché y memoria principal detectados, el tipo de monitor (normalmente, VGA) y la configuración para los puertos serie y paralelo.

- 8 **Autoconfiguración de dispositivos.** Nuestra BIOS PnP procede a la autoconfiguración de los dispositivos que dispongan de la facilidad PnP, realizando sobre ellos la secuencia de pasos descrita en la [sección 23.1.2.2](#), y anotando en el área ESCD de la RAM-CMOS la configuración final realizada.

En este área se dispone además de información que permite saber si el sistema ha variado en algún aspecto su configuración desde la última vez que fue arrancado. De no ser así, la

☛ [pág. 185](#)  
primer atajo

[pág. 45](#) ☛

despierta  
el monitor

listos  
los discos

[pág. 48](#) ☛

operativo  
el teclado

[p. 254/Vol. 1.2](#) ☛

pantalla de  
presentación

[pág. 38](#) ☛

inventario

tabla ASCII

PnP

[pág. 11](#) ☛

segundo atajo

BIOS sabe que no es necesario activar todo el protocolo de configuración, saltándose esta parte de la secuencia de arranque y limitándose a emitir un breve informe descriptivo de las prestaciones encontradas en los dispositivos alojados en los zócalos de expansión del sistema que completa a la tabla ASCII anterior.

De existir algún conflicto derivado de una mala configuración, esta información puede indicarnos la fuente del problema, aunque en ocasiones permanece en pantalla de una forma tan efímera que cuesta trabajo leerla con un mínimo de detalle.

⑨ **Iniciación del sistema operativo.** El programa inicial concluye su ejecución cargando en memoria el proceso *bootstrap* que procederá al arranque del sistema operativo. Para ello, activa el dispositivo (disco) en el que éste se encuentra almacenado, y procede a cargar en memoria el registro maestro de arranque (master boot record) ubicado de forma fija en el primer sector de datos del disco (al margen de que se trate del disco flexible, disco duro o CD-ROM, este sector es en todos los casos el primero del cilindro 0, pista 0 y cabeza 0). Este registro maestro no es más que la parte residente del sistema operativo, es decir, un primer embrión del que irán habilitándose los sucesivos módulos conforme vayan siendo necesarios.

arranca el S0

Existe una opción dentro del menú BIOS Features Setup (ver más adelante la [sección 24.3.3](#), opción **BOOT SEQUENCE**) en el programa de configuración del sistema, donde el usuario puede indicar a la BIOS el dispositivo en el que se encuentra almacenado el sistema operativo (normalmente, el disco duro o un CDROM). El usuario puede hacer uso de esta facilidad para evitar que la BIOS vaya mirando uno por uno en la lista de dispositivos candidatos según el orden que tenga preestablecido, ahorrándose la consiguiente demora.

← pág. 45

tercer atajo

⑩ **Transferencia de responsabilidades.** El programa de arranque finaliza transfiriendo el control al sistema operativo, que utilizando un agradable entorno gráfico como interfaz, quedará a disposición de las órdenes del usuario.

Si en lugar de encender el sistema pulsamos las teclas CTRL+ALT+DEL una vez está activo, se producirá la reiniciación del sistema, activándose una secuencia de arranque más corta. Esta secuencia se señalará con la señal **INIT** del procesador en lugar de con **RESET**, y presupone que los registros internos del procesador se encuentran ya inicializados y el programa inicial está cargado en memoria, por lo que retoma su ejecución en el paso 4, con las operaciones de autotesteo de dispositivos.

reiniciación del sistema

Esta reiniciación del sistema, conocida en el argot como arranque en caliente, produce la ejecución de una versión abreviada de las rutinas POST en la que, por ejemplo, se prescinde del chequeo extensivo de las celdas de memoria principal, en consonancia con lo que ocurre tras la activación de la opción **QUICK POWER ON SELF TEST** del menú **BIOS FEATURES SETUP**. Para conocer si se debe efectuar un arranque en frío (desde el principio de la secuencia de arranque) o en caliente, la BIOS comprueba el dato almacenado en la posición de memoria 472h, donde un valor 1234h señala la reiniciación en caliente, y cualquier otro valor, un arranque en frío.

arranque en caliente

 **Resumen** 

La Flash-BIOS y la RAM-CMOS constituyen el eje central para la configuración del PC en su más bajo nivel, enmarcándose en una capa intermedia entre el hardware y el software del sistema conocida como firmware.

firmware

La RAM-CMOS comenzó proporcionando celdas de memoria no volátil de forma separada, para posteriormente integrarse en el puente sur del juego de chips a partir de la quinta generación. La BIOS, en cambio, ha tenido siempre su chip propio, incorporando en su evolución tecnológica

evolución

Característica	Flash-BIOS	RAM-CMOS
Chip que la implementa	Propio	Puente sur de la placa base
Formato	DIP	BGA
Desmontable	SI	NO
Información que almacena	Programa de configuración	Parámetros de configuración
Capacidad	128 Kbytes - 1 Mbyte	64 - 256 bytes
Consumo de energía	3-3.3 v. / 0.5 A.	3-3.3 v. / 1 $\mu$ A
Suministro de energía	Discontinuo	Permanente
Procedencia del suministro	Fuente de alimentación	Pila de litio
Evolución tecnológica	ROM, EPROM, Flash	RAM no volátil
Principales fabricantes en el entorno PC	Award, MicroId, American Megatrends (AMI)	Dallas Semic., BenchmarkQ, ST Microelectronics

TABLA 23.2: Los diez rasgos más importantes de la Flash-BIOS y la RAM-CMOS.

la facilidad de autoconfiguración (PnP) y actualización (Flash).

La [tabla 23.2](#) sintetiza las principales cualidades de cada una de ellas. La Flash-BIOS es la responsable del código del firmware, y la RAM-CMOS, de sus datos asociados, estando ambas íntimamente relacionadas con el tratamiento de interrupciones que gobierna el funcionamiento del PC.

Adicionalmente, la BIOS suministra un programa para el arranque del PC y otro programa para la configuración del sistema mediante un sellillo y claro interfaz. En el primero se encuentra la clave para entender nuestro proceder en el [capítulo 28](#) dedicado a la detección y reparación de averías en el PC, y en el segundo están reunidos todos los secretos para su adecuada configuración que desmenuzaremos en el [capítulo 24](#).

pág. 165 ➔

pág. 35 ➔



## La anécdota: Flash, esa pequeña gran protagonista



Resulta difícil encontrar en la actualidad algún dispositivo tecnológico relacionado con el PC en el que éste no constituya su primer mercado de destino. Pero si son pocos los que han conseguido librarse de semejante yugo, entre ellos sobresale un elemento que se ha ganado una creciente autonomía por derecho propio: La memoria de tipo Flash.

A caballo entre las memorias de semiconductores (memoria principal y caché) y las memorias de almacenamiento magnético (disco duro), este tipo de memoria conjuga la velocidad de acceso de la primera con la perpetua retención de datos del segundo.

Las prestaciones de las memorias Flash fueron inicialmente aprovechadas dentro del PC para dotar al programa de configuración del sistema de un carácter dinámico que le permitiese actualizar sus contenidos como forma de presentar cierta resistencia frente a su pertinaz envejecimiento. Empresas como Transmeta utilizaron esa tecnología para conseguir ese mismo grado de metamorfosis en el corazón de procesadores como el Crusoe, y de paso, aprovechar el bajo consumo de sus productos para orientarlos al segmento de los portátiles.

Pero esa tecnología abrió todo un mundo de posibilidades que pronto supo ser aprovechada para mejorar la funcionalidad de otro tipo de dispositivos, pequeños en su tamaño y ágiles en su evolución. Los teléfonos móviles, las cámaras fotográficas digitales y los reproductores de MP3 son los tres ejemplos más destacados. Todo este ejército de pequeños electrodomésticos de consumo tienen un denomina-

otras aplicaciones