

Unidad de Trabajo 3: El Microprocesador.

1. Concepto y evolución
2. Estructura y componentes
3. Funcionamiento.
4. Características y medidas.
 - 4.1. Frecuencia de reloj.
 - 4.2. Tecnología de integración.
 - 4.3. Paralelismo a nivel de instrucción
 - 4.3.1. Etapas básicas en las que se divide el proceso de ejecución de una instrucción.
 - 4.3.2. Segmentación.
 - 4.3.3. Superescalaridad.
 - 4.4. Memoria caché integrada
 - 4.5. Conjunto de instrucciones que maneja
 - 4.5.1.

Evolución de los microprocesadores.

MODELO	AÑO	BUS DATOS	VELOCIDAD EXTERNA	VELOCIDAD INTERNA
8088	1979	8bits	4,77 MHz	14 MHz
80286	1982	16 bits		15 MHz
80386	1985	32 bits		20 MHz
80486	1989	32 bits		25 MHz
Pentium 1	1993	64 bits		60 MHz
Pentium pro	1995	64 bits	66 MHz	200 MHz
Pentium II	1997	64 bits	66-100 MHz	266 MHz
Pentium III	1999	64 bits		550 MHz
Pentium IV	2001	64 bits	400 MHz	2 GHz

- El primer PC lo creó IBM en los 70. Todos los micros se han ido basando en el diseño del 8088.
- El tener 32 bits permite tener un software más moderno.
- El 486 es el primer micro que incorpora el coprocesador matemático y también la mejora que incorpora la memoria caché L1.
- En el 95 pasamos al Pentium Pro donde se trabaja la velocidad interna y externa dando un gran salto.

Grandes avances.

- Soporte para la memoria virtual, con el 286, el micro puede utilizar parte del disco duro como una memoria RAM.
- En el 486 se incorporó el coprocesador al micro, la gran ventaja era que las operaciones matemáticas se hacían más deprisa.
- Soporte de sistemas multiprocesador.

Procesadores en el mercado.

- AMD: www.amd.com
 - ATHLON 64 Fx (juegos y medios digitales)
 - ATHLON 64 X2 de doble núcleo
 - TURION 64 X2 Dual-Core (portátiles)
 - OPTERON (servidores).
- Intel: www.intel.com
 - Core™2 Duo
 - Core™2 Extreme (juegos y multimedia)
 - Intel® Pentium® con tecnología de procesador de doble núcleo (portátiles).

El futuro.

Hace tiempo que se oye hablar de otras alternativas a los microprocesadores tal y como se conocen en la actualidad. Desde los micros biológicos hasta la computación molecular rondan los sistemas microinformáticos hace tiempo, previendo los límites físicos de integración del silicio. Ciertamente se ha llegado a tal punto que pueden cuantificarse los átomos empleados para realizar determinados procesos de fabricación y, asumiendo la indivisibilidad del átomo (no tan asumida, vaya), parece que necesariamente deberá producirse un cambio en cuestión de no demasiados años. De cualquier manera, la realidad abre aún un abanico de opciones interesantes y todavía queda tiempo para seguir concibiendo los microprocesadores tal y como se hace en la actualidad, basados en semiconductores. El tiempo, sin duda, deparará nuevas sorpresas.

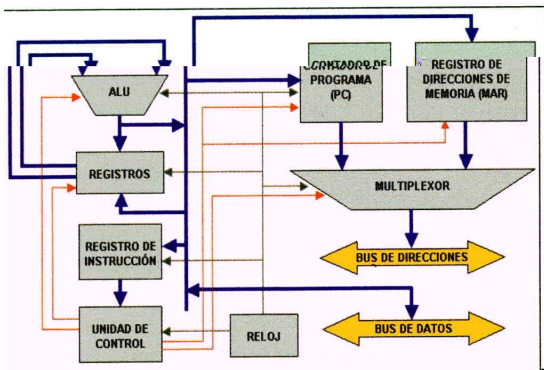


Figura 1.
Estructura de un procesador sencillo. Se pueden apreciar los componentes básicos de cualquier procesador.

2.- Estructura básica de la CPU

La Figura 1 muestra la estructura interna de un procesador sencillo con arquitectura Von Neumann. El diseño es básico (se corresponde con el presentado por algunos procesadores sencillos de 8 bits) pero su comprensión permitirá entender el funcionamiento de procesadores más complejos, que amplían esta estructura.

Se pueden apreciar los siguientes elementos:

Bus de direcciones. Permite enviar direcciones a la memoria y periféricos. Las direcciones son números que indican la posición de memoria donde se desea escribir un dato, o de donde se desea leerlo. La memoria se abordará con detalle en próximos capítulos, al igual que los periféricos.

Bus de datos. Permite al procesador enviar datos a la memoria y periféricos. De la misma forma, permite que el procesador reciba datos de dichos elementos. Para escribir (o leer) un dato, primero es necesario colocar la dirección donde escribirlo (o de donde leerlo) en el bus de direcciones.

Registro de instrucción. En cada momento, almacena la instrucción que está siendo ejecutada por la CPU.

Archivo de registros. Son almacenes temporales de datos, de acceso muy rápido. En general, almacenan los operandos sobre los que actúan

las instrucciones, y también los resultados de éstas (para un futuro uso). También existen registros de aplicación específica, como ya veremos.

Unidad Aritmético-Lógica. Más conocida por el acrónimo inglés ALU (*Arithmetic-Logic Unit*). La ALU es el motor de cálculo del procesador, ya que se encarga de realizar las operaciones para las que está capacitado. Como se aprecia en la Figura 1, la ALU es capaz de tomar 2 datos como operandos y producir una salida, resultado de aplicar una operación (seleccionable mediante unas líneas de control). Las operaciones soportadas pueden ser aritméticas (suma, resta, etc.), lógicas (and, or, etc.) y otras operaciones como desplazamientos de bits. Sin duda, la ALU es un elemento muy importante, ya que define las operaciones de cálculo que la CPU puede realizar.

Contador de programa. Se trata de un registro especial, al que se denomina normalmente PC (*del inglés Program Counter*). El PC contiene, en cada momento, la dirección de memoria en la que se encuentra la siguiente instrucción a ejecutar. Cada vez que se obtiene una nueva instrucción de la memoria, el PC se actualiza para apuntar a la próxima instrucción. Normalmente, dicha actualización consiste en un simple incremento, para apuntar a la celda de memoria siguiente (tenga en cuenta que, en general, las instrucciones de un programa se almacenan en posiciones de memoria consecutivas). Este comportamiento cambia cuando se eje-

cuta una instrucción de salto. En ese caso, el PC se actualiza con la dirección de destino del salto, donde se encuentra la siguiente instrucción a ejecutar.

Registro de direcciones de memoria. También denominado MAR (*Memory Address Register*). Este registro almacena la próxima dirección de memoria de la que leer (o en la que escribir) un dato. Se trata de un registro que complementa al PC: mientras que este último apunta a instrucciones, el MAR apunta a datos.

Multiplexor. No se trata de un elemento exclusivo de las CPU, sino de una pieza fundamental del diseño de sistemas digitales. En el esquema de la Figura 1, el multiplexor actúa como una especie de conmutador: transporta el contenido del PC o del MAR hacia el bus de direcciones. La línea roja de la parte izquierda es el terminal que permite seleccionar si es el contenido del PC o el del MAR el que llegará al bus (colocando en él un 1 o un 0 lógico). En la Figura 2 se ilustra este hecho.

Reloj. En la actualidad, prácticamente todos los procesadores son sistemas digitales síncronos. Esto significa que trabajan al ritmo marcado por una señal de reloj que no es más que un conjunto de pulsos distanciados por igual en el tiempo. Más adelante veremos que una mayor velocidad de reloj no siempre implica un mayor rendimiento del procesador.

Unidad de control. Es el bloque encargado de coordinar el funcionamiento interno de la CPU. Indica a cada componente interno cómo debe funcionar y cuándo tiene permiso para entrar en acción. En otras palabras, se puede entender como el director de orquesta de la CPU.

Con los anteriores comentarios se ha obtenido una visión de sistema de la CPU. Como en todo sistema, se dispone de bloques que cooperan entre sí para lograr una labor común (en este caso, ejecutar programas).

3.- Funcionamiento

Considere una posible instrucción a ejecutar por el procesador de la Figura 1 (denominada, por ejemplo, "ADD R1, R2, R3"). Suponga que dicha instrucción toma los datos almacenados en los registros denominados R2 y R3, realiza la suma y almacena el resultado en el registro R1. Como ya habrá intuido, R1, R2 y R3 se encuentran en el archivo de registros. El proceso completo para ejecutar dicha instrucción es el siguiente:

- 1.- El Contador de Programa (CP) se actualiza con la dirección de memoria donde se halla la siguiente instrucción a ejecutar ("ADD R1, R2, R3").
- 2.- La unidad de control (UC) modifica el estado del multiplexor, de forma que el contenido del CP se dirija hacia el bus de direcciones.
- 3.- La memoria recibe la dirección contenida en el CP, y responde devolviendo el dato contenido en dicha celda a través del bus de datos. Dicho dato representa a la instrucción "ADD R1, R2, R3" mediante un valor numérico binario.
- 4.- La instrucción recibida se almacena en el registro de instrucción.
- 5.- La UC descodifica la instrucción. Es decir, averigua cuál es la instrucción (ADD) y cuáles son los operandos sobre los que ésta trabaja (R1, R2 y R3). Se debe tener presente que una instrucción se representa por un número binario, que ofrece información sobre la operación a realizar (mediante un grupo de bits), y los operandos sobre los que trabajar (en otros grupos de bits). Para facilitar su comprensión, las instrucciones se suelen etiquetar mediante mnemónicos (por ejemplo, "ADD"), formando lo que se denomina "lenguaje ensamblador", propio de cada procesador.
- 6.- La UC se comunica con el archivo de registros (véase la conexión en rojo) y hace que el contenido de los registros R2 y R3 vaya a parar a las 2 entradas de la ALU.
- 7.- La UC configura la ALU (mediante la línea de control en rojo, que realmente son varias líneas) indicando que se desea realizar la suma.
- 8.- La UC configura el archivo de registros, de forma que el resultado de la suma se dirija hacia el registro R1.
- 9.- La UC ordena a la ALU que realice la operación configurada. Tal y como se pretendía, en el registro R1 se dispone de la suma de los valores contenidos en R2 y R3, con lo que la instrucción ha sido completada.

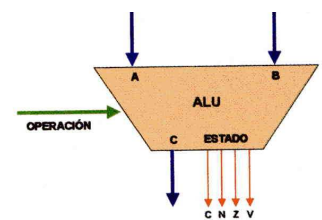


Figura 3. Estructura general de una Unidad Aritmético-Lógica (ALU).

Como conclusión importante del ejemplo, conviene remarcar que la ejecución de una instrucción sencilla se compone de una serie de operaciones internas a la CPU, todas controladas por la UC. Esta última sabe cuál es la secuencia de tareas a aplicar para cada instrucción, tras la etapa de descodificación.

4.- Características y medidas

El microprocesador es extremadamente complejo, por lo que lo primero que debemos aprender es a reconocer los parámetros que realmente influyen en su rendimiento:

4.1.- Frecuencia de reloj

Indica la velocidad a la que trabaja el microprocesador. Aunque se ha extendido la idea que especifica el número de instrucciones que puede ejecutar por segundo, no es del todo cierto. Veamos realmente lo que quiere decir este parámetro.

El reloj es un pequeño cristal de cuarzo, ubicado en la placa base, que proporciona una señal de onda digital, binaria, cuadrada, síncrona y periódica.

Esta señal sincroniza todas las operaciones que se realizan en cada uno de los dispositivos del ordenador, por lo que también sincroniza las operaciones que se ejecutan en el microprocesador.

El **período** o ciclo de reloj es la duración de un pulso u oscilación del reloj. Por el contrario, la **frecuencia** expresa el número de ciclos producidos por el reloj en una unidad de tiempo, por lo que puede expresarse como la inversa del período. Las unidades utilizadas para medir la frecuencia son los hertzios (Hz) y determinan el número de ciclos por segundo. Esta unidad puede resultar pequeña, por lo que disponemos de los siguientes múltiplos:

Megahertzios (MHz): $1 \text{ MHz} = 1.000.000 \text{ Hz}$.

Gigahertzios (GHz): $1 \text{ GHz} = 1.000 \text{ MHz} = 1.000.000.000 \text{ Hz}$.

Debido a que nuestro sistema informático incorpora dispositivos muy dispares, que operan a diferentes velocidades, existen multiplicadores y divisores de frecuencia, generando, de esta manera, señales con frecuencias distintas, destinadas a sincronizar las operaciones tanto en los dispositivos más lentos como en los más rápidos.

El período es la unidad básica de tiempo. Cualquier operación debe consumir un número entero de períodos para que puedan sincronizarse. Normalmente, cada instrucción que se ejecuta en el microprocesador utiliza varios ciclos de reloj, ya que se descomponen en operaciones atómicas que consumen un período de reloj cada una de ellas. Por consiguiente, cuanto mayor sea la frecuencia del microprocesador, más rápido ejecutará las instrucciones y menor tiempo invertirá en finalizarlas.

Intel ya supera los 3 GHz con sus Pentium 4.

4.2.- Tecnología de integración

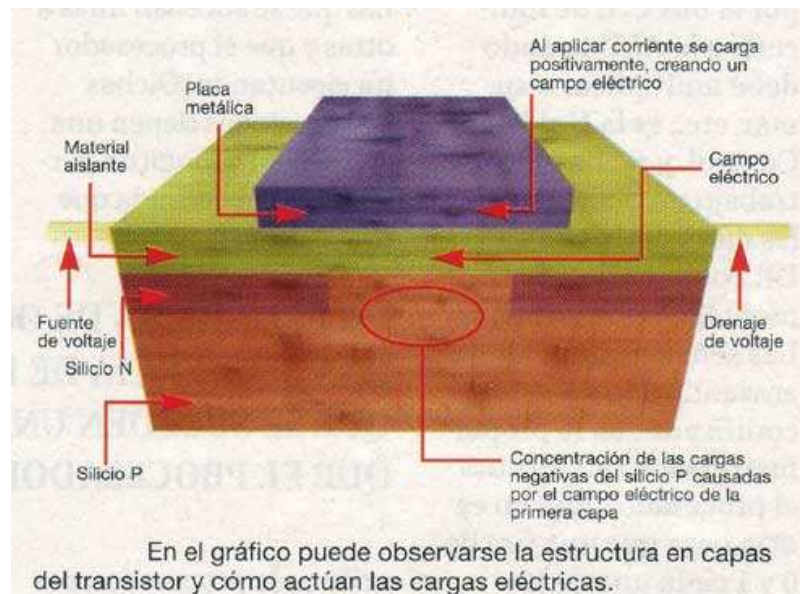
El microprocesador es un conjunto de componentes que procesan una señal eléctrica que puede tomar dos estados: ausencia o presencia de corriente (0 y 1 = lenguaje binario). De esta forma, todo programa es traducido a este lenguaje para que pueda ser procesado.

El microprocesador está compuesto por millones de **transistores** que son minúsculos interruptores que permiten o niegan el paso de corriente. Combinando estos elementos se pueden construir estructuras más complejas para construir puertas lógicas (AND, OR, NOT, XOR...) y circuitos integrados.

Dentro de un microprocesador encontramos millones de transistores. La integración de estos componentes en el área del chip tiene lugar mediante sofisticadas técnicas en el que los transistores se disponen en una estructura bidimensional que ocupa la capa inferior y se interconectan mediante un enrejado compuesto por capas de aluminio o cobre que se sitúan por encima de los transistores. El material que forma dichos transistores es el silicio.

Pero el valor que nos interesa en este apartado es la **distancia de integración**, que viene determinada por la separación de los dos bloques tipo n situados en la capa semiconductor y cuya medida viene expresada en micras (millonésima parte del metro). Conforme avanza la tecnología se tiende a disminuir la distancia de integración, proporcionando los siguientes beneficios:

- Los transistores construidos son más pequeños, por lo que podemos integrar mayor número en el mismo espacio. Esto conlleva un aumento de las prestaciones del microprocesador, como la incorporación de nuevas unidades funcionales o el aumento de las memorias caché.
- Aumenta la velocidad de operación del transistor, ya que al ser menor la distancia que tiene que recorrer el flujo eléctrico también es menor el tiempo que tarda en recorrerla. Esto conlleva un aumento de la frecuencia debido a que las operaciones se efectúan más rápidamente.
- Disminuye el voltaje que necesita el microprocesador para ser alimentado. De esta forma se consigue una disminución del calor disipado pudiendo aumentar la frecuencia.
- Disminuye el coste de fabricación, ya que la cantidad de silicio empleada es menor.



La distancia de integración ha evolucionado desde 1 micra de los 486, hasta las 0,13 micras de los actuales Pentium 4 y Athlon XP, pasando por las 0,35 micras de los Pentium II o las 0,25 de los Pentium III.

El número de transistores que forman un 486 es de 1.200.000, en un Pentium encontramos 3.100.000, en un Pentium II → 7.500.000, en un Pentium III → 28.100.000 y 55.000.000 en un Pentium 4.

4.3.- Paralelismo a nivel de instrucción

Mediante esta característica se intenta eliminar la ejecución secuencial de instrucciones, posibilitando la ejecución simultánea. Para ello se aprovecha la existencia de las distintas unidades funcionales del microprocesador, ya que una misma instrucción no puede trabajar simultáneamente sobre varias unidades funcionales. Por ello, cuando una instrucción está haciendo uso de una unidad funcional, otra instrucción puede trabajar simultáneamente en otra unidad funcional. Veamos las distintas

formas en que puede ser tratado el paralelismo: Segmentación, superescalaridad y supersegmentación.

- **Segmentación o pipelining:**

Con esta técnica cada instrucción que entra a ejecutarse en el microprocesador se divide en una serie de etapas. Utilicemos, por ejemplo, el siguiente modelo de cinco etapas:

- Etapa 1: Búsqueda de la instrucción (E1).
- Etapa 2: Decodificación (E2).
- Etapa 3: Lectura de operandos (E3).
- Etapa 4: Ejecución (E4).
- Etapa 5: Escritura del resultado (E5).

De esta manera, mientras una instrucción está ejecutando su etapa 1, otra puede ejecutar su etapa 2, una tercera puede ejecutar su etapa 3 y así sucesivamente. Destaquemos que cada instrucción se ejecutará secuencialmente de principio a fin, pero solapándose con otras instrucciones que también se ejecutan secuencialmente en el resto de unidades funcionales.

El número de etapas en la segmentación varía según el modelo de microprocesador. En la actualidad pueden llegar a las 20 etapas.

- **Superescalaridad:**

Ya hemos visto que las actuales técnicas de integración permiten incluir más componentes en el área del microprocesador. Es por ello que también se han efectuado réplicas de unidades funcionales para que varias instrucciones puedan ejecutar simultáneamente en una misma etapa. En la actualidad el factor de superescalaridad se encuentra entre 3 y 5, es decir, si este factor es 5, lo que quiere decir es que un total de 5 instrucciones pueden ejecutar de forma simultánea en la misma etapa, ya que existen 5 unidades funcionales disponibles para poder ejecutar cada una de ellas. Utilizando el modelo de 5 etapas visto anteriormente podemos mostrar gráficamente la evolución en la ejecución de un sistema con factor de superescalaridad 3 en la siguiente figura.

- **Supersegmentación:**

La combinación de las dos técnicas anteriores producen microprocesadores supersegmentados. La gráfica adjunta muestra la evolución de un sistema con factor de supersegmentación 2. Observemos que el período del reloj ha disminuido en un factor determinado por el nivel de supersegmentación, en este caso 2; es decir, los ciclos de reloj son la mitad que en los casos anteriores. Esto conlleva a la construcción de microprocesadores con frecuencias muy elevadas.



Inconvenientes del paralelismo: las dependencias:

Hasta ahora hemos mostrado la posibilidad de ejecutar instrucciones de un mismo programa de forma simultánea, pero esto choca con la propia estructura secuencial que poseen los comandos de las aplicaciones. Veamos los riesgos asociados a las técnicas de paralelismo:

- **Dependencias de datos:** Si tenemos dos instrucciones en que la segunda necesita el resultado de la primera, la segunda no podrá comenzar su ejecución hasta que no haya finalizado la primera.
- **Dependencias de control:** Una instrucción de salto condicional interrumpe la ejecución concurrente de instrucciones, ya que hasta que no sea evaluada la condición no conoceremos la siguiente instrucción a ejecutar.

Soluciones:

- **Ejecución fuera de orden:** Cuando una instrucción no puede ejecutarse por dependencia de datos el microprocesador continúa con la siguiente.

Predicción de salto: Una instrucción de salto condicional produce dos caminos por los que el programa puede continuar su ejecución. Para eliminar las dependencias de control el microprocesador elige uno de estos caminos para continuar la ejecución. Si acierta habremos avanzado en la ejecución de la aplicación, si falla simplemente tendremos que comenzar la ejecución por el otro camino.

4.4.- Memoria caché integrada

Cuando uno afronta la compra de un microprocesador, su caché es uno de los puntos clave en los que debe fijarse. En los microprocesadores actuales existen dos cachés integradas dentro del chip del microprocesador (aunque son realmente tres). Éstas son:

- Caché de primer nivel (L1): Suele estar separada en dos: Una para datos y otra para instrucciones.
- Caché de segundo nivel (L2): Una caché más grande pero más lenta que la anterior, aunque funcione a la misma velocidad que la anterior. Almacena a la vez datos e instrucciones.

Las cachés integradas funcionan a la misma velocidad que el microprocesador.

Actualmente, la caché L2 está integrada en el microprocesador pero en el Pentium II, una versión del Pentium III y otra del AMD K7, dicha memoria era interna, es decir era un chip aparte colocado en el microprocesador. Las cachés internas funcionan a la mitad o a 1/3 de la velocidad del microprocesador.

La caché L1 empezó a integrarse dentro del microprocesador con el Pentium. La caché L1 es más pequeña que la L2. Así por ejemplo, hablaremos de cachés L1 de datos o de instrucciones de 32 Kbytes, 16 Kbytes... Para la caché L2 encontramos tamaños de 256 Kbytes (integrada), 512 kbytes (interna)...

El microprocesador siempre busca los datos y las instrucciones en la caché L1. Si aquí no los encuentra, se accede a la caché L2. Si están en L2, se cargan en L1 y se envían al microprocesador. Si no estuviesen tampoco en L2, se accede a la memoria principal y se cargan dentro de L2, luego a la L1 y se envían después al microprocesador.

- Bus frontal: Comunica memoria principal con caché L2.

- Bus trasero: Comunica caché L2 con caché L1.

4.5.- Conjunto de instrucciones que es capaz de entender el microprocesador

Atendiendo al número y complejidad de instrucciones que son capaces de entender, los microprocesadores pueden ser clasificados como CISC y RISC.

Los **CISC** son aquellos que utilizan un conjunto complejo de instrucciones (de 200 a 500) y los **RISC** son los que usan un conjunto reducido de instrucciones simples (128 o menos). En un ordenador RISC serán más grandes los programas ejecutables (contendrán un mayor número de instrucciones) pero se reducirá el tiempo que tarda en ejecutarse cada instrucción, ya que son más simples que en los CISC.

La tendencia actual es dotar a los microprocesadores cada vez de más características de tipo RISC; Los RISC suelen incorporar los últimos avances; y además suelen ser más rápidos, tener menos transistores, y por tanto, ocupan menos superficie en silicio. Esto hace que sean más baratos de producir que los CISC, aunque también importa el volumen de producción a la hora de establecer un precio.

Incluso se empieza a hablar de híbridos, que, aunque utilizan un conjunto complejo de instrucciones, internamente son RISC. Esto se debe a la necesidad de utilizar una arquitectura heredada, aumentando al mismo tiempo las prestaciones.

Un microprocesador de tipo RISC, en contraposición al tipo CISC, tiene alguna o todas las características siguientes:

- Conjunto reducido de instrucciones. Además, las instrucciones suelen ocupar todas el mismo número de bytes (1 palabra de memoria).
- Conjunto reducido de modos de direccionamiento, que a veces se reduce a sólo 1, direccionamiento directo.
- 1 instrucción, 1 ciclo: la mayoría de las instrucciones más usuales están optimizadas, para ejecutarse en un sólo ciclo. Los CISC necesitan varios ciclos para ejecutar cada una de sus instrucciones. Si establecemos una comparación, los RISC ejecutan varias instrucciones simples y muy rápidas por cada instrucción compleja de los CISC. Los CISC no suelen utilizar microcódigo, o sea, la unidad de control suele estar *cableada*.
- Muchos registros: tienen muchos más registros que un CISC, y los manejan por bancos de registros (por ejemplo, de 16 en 16 registros), de forma que cada programa puede tener su propio grupo de registros asignado, en un entorno multiprograma. El conjunto de todos los registros se llama fichero de registros o *register file*. Los registros son, además, de propósito general: no hay un contador específico, ni un registro de pila específico.
- Trabajo intensivo con los registros: casi todas las operaciones aritméticas y lógicas trabajan sobre los registros, de forma que para operar con un dato, hay que transferirlo previamente al banco de registros. Para ello, hay también instrucciones rápidas que mueven bloques de memoria al banco de registros. Esos bloques pueden proceder de la caché.
- Pipeline: de forma que, en cada ciclo, pueden terminar varias instrucciones simultáneamente. En este caso, se llaman *superescalares*, son capaces de ejecutar más de una instrucción por ciclo. Lo más habitual es que los RISC o pseudoRISC tengan varios pipelines, incluso uno de coma flotante y dos de enteros. Los algoritmos necesarios para mantener los pipelines llenos la mayor parte del tiempo son bastante complejos.
- Arquitectura tipo Harvard: el bus de datos y el de instrucciones están separados, cada uno llega al procesador por caminos diferentes, y por tanto se procesan en paralelo.
- Compiladores complejos: debido a lo anterior, para sacar un máximo partido, los compiladores deben de optimizar el código de forma que los pipelines estén llenos durante el mayor tiempo posible. Lo más habitual es que no se programen en código máquina ni ensamblador.

Normalmente los microprocesadores se agrupan en *familias*, cuyos miembros son compatibles entre sí, en el sentido que los miembros más modernos de la familia están diseñados para entender el código máquina de los miembros más antiguos (lo cual no garantiza que el programa funcione, claro está). Esto es lo que se suele denominar *compatibilidad binaria*, y también *compatibilidad ascendente*. Eso garantiza que existan muchas aplicaciones para los nuevos microprocesadores en el momento que salen al mercado. Aparte, cada microprocesador nuevo añade alguna característica (a menudo sólo una mayor velocidad), de forma que, habitualmente, no son compatibles en pines con los modelos más antiguos; incluso aunque tengan el mismo número de pines, estos tienen funciones diferentes. O sea que, en la mayoría de los casos, no se puede simplemente despegar un microprocesador y pegar otro.