

Componentes, estructura y
funcionamiento de la Unidad
Central de Proceso.

TEMA 3

ABACUS NT

Índice

1. Introducción

2. Principios funcionales de la CPU

2.1. Arquitectura de Von Neumann versus Arquitectura Harvard

- 2.1.1. Arquitectura de Von Neumann
- 2.1.2. Arquitectura Harvard
- 2.1.3. Taxonomía de Flynn

3. Componentes y estructura de la CPU

3.1. Unidad de Control

3.2. Unidad Aritmético-Lógica

3.3. Unidades Funcionales de las CPU Actuales

- 3.3.1. Memoria Caché
- 3.3.2. Coprocesador matemático (FPU)
- 3.3.3. Unidad de gestión de Memoria (MMU)
- 3.3.4. Unidad de predicción de salto (Branch Predictor)
- 3.3.5. Unidad de instrucciones extendidas
- 3.3.6. Núcleos especializados y SoC

3.4. Arquitectura del Sistema de Entrada/Salida

- 3.4.1. Bus de datos:
- 3.4.2. Bus de direcciones:
- 3.4.3. Bus de control:
- 3.4.4. Buses multiplexados:
- 3.4.5. Periféricos.

4. Funcionamiento de la unidad central de proceso.

4.1. Fases de ejecución de las instrucciones.

- 4.1.1. Juego de instrucciones.

5. Ampliación

5.1. Formato de las instrucciones.

5.2. Modos de funcionamiento de la CPU

- 5.2.1. Aritmética de segmento
- 5.2.2. Modo Real
- 5.2.3. Modo Protegido

5.3. CISC versus RISC

- 5.3.1. Tecnología CISC
- 5.3.2. Tecnología RISC

6. Conclusión

6.1. Relación del tema con el sistema educativo actual

7. Bibliografía

1. Introducción

Un **computador, computadora u ordenador**, es una máquina electrónica capaz de aceptar unos datos de entrada, efectuar con ellos operaciones lógicas y aritméticas y proporcionar la información resultante a través de un medio de salida; todo ello de forma automática, es decir, sin necesidad de la intervención de un operador humano, y bajo el control de un programa previamente almacenado en el propio ordenador.

Es importante considerar que este flujo de información se realiza de forma digital por lo general. El término dígito (etimológicamente deriva de dedo en latín) nos da una idea aproximada de que la representación no va a tomar cualquier valor del mundo real (señales analógicas) sino sólo un número discreto de estos al igual que el número de dedos de nuestras manos es limitado, al igual que el sistema decimal (base diez).

En el presente tema vamos a estudiar cómo un ordenador realiza esos procesos, con qué componentes precisos y bajo qué principios de funcionamiento.

2. Principios funcionales de la CPU

En este tema vamos a abordar la **arquitectura** de un ordenador según los **niveles conceptuales de Blaauw**, es decir su **comportamiento funcional**.

2.1. Arquitectura de Von Neumann versus Arquitectura Harvard

2.1.1. Arquitectura de Von Neumann

El modelo básico de arquitectura empleada en los ordenadores digitales fue establecido en 1945 por John Von Neumann, físico húngaro afincado en Estados Unidos.

Su gran aportación con respecto a las arquitecturas preexistentes (arquitectura Harvard principalmente) fue eliminar la **lógica de programación basada en relés o conexiones cableadas**, unificando memoria de datos y memoria de programa en una **única memoria principal** y simplificando por consiguiente la doble gestión de control de datos y programa. De esta forma además se garantiza la **disponibilidad incremental de espacio** en memoria ya que este se puede dividir entre datos y programas a voluntad.

Los **elementos funcionales** de la arquitectura Von Neumann son, por tanto:

Unidad de Control

Unidad Aritmético-lógica

Memoria Principal

Buses de Entrada/Salida

La unidad de control y la unidad aritmético-lógica se agrupan en la denominada **unidad central de proceso**. Más adelante desarrollaremos estos elementos.

2.1.2. Arquitectura Harvard

El término proviene del diseño de la computadora **Mark I basada en relés** y desarrollada en la universidad de Harvard.

La característica diferenciada de esta arquitectura reside en que **la memoria de datos y de instrucciones está separada**, así como la lógica de control de las mismas.

En la arquitectura de Harvard original, no existe realmente una memoria de programa, sino que **la programación se realiza mediante el encendido y apagado de relés**, ya que la idea de considerar la programación en términos de memoria fue posterior.

Esta arquitectura permite leer en paralelo datos e instrucciones, porque cuenta con un **sistema doble y diferenciado** de flujo y control.

Actualmente las arquitecturas **ARM, x86 y AMD64** son de tipo “**Harvard modificado**” ya que permiten acceso simultáneo a **memoria de programa y a memoria de datos al estar separada pero sólo en caché**, mientras que en la memoria principal se encuentran juntas al clásico estilo de la arquitectura Von Neumann.

Este tipo de arquitectura **Harvard Modificada** aúna las ventajas de ambas:

Mantienen una única **memoria principal de datos e instrucciones**, de forma que se simplifica la lógica de circuitos y la disponibilidad de espacio.

Permiten la protección de instrucciones a nivel de procesador mediante la separación a nivel de caché.

Al estar separados en la CPU, existe un **acceso paralelo y simultáneo a datos e instrucciones**, con el consiguiente incremento de rendimiento.

2.1.3. Taxonomía de Flynn

La taxonomía de Flynn es una clasificación de arquitecturas de computadores propuesta por Michael J. Flynn en 1972.

Clasificaciones

Las cuatro clasificaciones definidas por Flynn se basan en el número de instrucciones concurrentes (control) y en los flujos de datos disponibles en la arquitectura:

- Una instrucción, un dato (SISD)

Computador secuencial que no explota el paralelismo en las instrucciones ni en flujos de datos. Es la Arquitectura Von-Neumann. Un único procesador ejecuta un solo flujo de instrucciones para

operar datos en una única memoria. Se ejecuta una única instrucción y un dato en cada ciclo de reloj. Puede utilizar técnicas de segmentación o de pipelining. Ejemplos de arquitecturas SISD son las máquinas con uniprocesador o monoprocesador tradicionales como el PC o los antiguos mainframe.

- Una instrucción, múltiples datos (SIMD)

Todas las unidades ejecutan la misma instrucción sincronizadamente, pero con datos distintos. Es un computador que explota varios flujos de datos dentro de un único flujo de instrucciones para realizar operaciones que pueden ser paralelizadas de manera natural. Puede ser de Arquitectura vectorial o Arquitectura matricial.

- Múltiples instrucciones, un dato (MISD)

Poco común debido al hecho de que la efectividad de los múltiples flujos de instrucciones suele precisar de múltiples flujos de datos. Sin embargo, este tipo se usa en situaciones de paralelismo redundante, como por ejemplo en navegación aérea, donde se necesitan varios sistemas de respaldo en caso de que uno falle.

- Múltiples instrucciones, múltiples datos (MIMD)

Varios procesadores autónomos que ejecutan simultáneamente instrucciones diferentes sobre datos diferentes. Los sistemas distribuidos suelen clasificarse como arquitecturas MIMD; bien sea explotando un único espacio compartido de memoria, o uno distribuido.

3. Componentes y estructura de la CPU

La unidad central de Proceso (CPU = Central Process Unit) unifica los elementos funcionales de Unidad de Control y Unidad Aritmético-Lógica.

3.1. Unidad de Control

Es el centro neurálgico del ordenador, donde se procesan los programas y su función consiste en realizar la **búsqueda, carga, secuenciación y ejecución** de las instrucciones.

Estas instrucciones están escritas en **Lenguaje Máquina** y están formadas por un **código de operación**, que indica a la UC qué operación debe realizarse, y por hasta **dos datos o direcciones de memoria** que indican la localización de esos datos.

Para realizar su trabajo, la UC utiliza los siguientes elementos:

Registros, entre los que cabe destacar:

Contador de programa. Contiene en todo momento la dirección de memoria de la siguiente instrucción a ejecutar.

Registro de instrucción. Almacena la instrucción en curso, que llevará consigo el código de operación y en su caso, los operandos o las direcciones de memoria de los mismos.

Registro de direcciones de memoria Se utilizan para almacenar direcciones de memoria que contienen los datos de una instrucción o el lugar donde almacenar el resultado de la misma.

Registro de estado. Contiene información sobre el resultado de la operación anterior de la unidad aritmético lógica y de posibles situaciones anómalas, como desbordamientos, interrupciones, etc.

Otros registros, como los **registros genéricos** que pueden ser utilizados por los propios programas y otros **registros específicos** utilizados por el procesador (**puntero de pila, registro de base**, etc.) y que dependen de su arquitectura concreta.

Reloj. El reloj es un circuito oscilador que genera autónomamente una señal en forma de pulsos a intervalos constantes que marcan los instantes en que han de comenzar los distintos pasos de que consta cada instrucción.

Decodificador. Se encarga de extraer el código de operación de la instrucción en curso, analizarlo y emitir las señales necesarias al resto de elementos para su ejecución a través del secuenciador.

Secuenciador. Genera una serie de órdenes muy elementales que permiten ejecutar la instrucción en curso, de forma síncrona con el reloj del sistema.

Por último, cabe señalar que las Unidades de Control se pueden clasificar en **dos** Categorías:

UC cableadas. La lógica está implementada en el hardware, lo que las convierte en UC de **propósito específico** y no es modificable.

UC micropogramadas. En ellas, la lógica se implementa como un programa almacenado como **firmware**, lo que permite su posterior reescritura.

3.2. Unidad Aritmético-Lógica

Es la unidad funcional encargada de realizar las operaciones aritméticas y lógicas **bajo la supervisión de la unidad de control**.

Las operaciones que puede efectuar esta unidad **solían ser muy elementales**, puesto que normalmente estaban formadas por un circuito sumador-restador, y basaban todas las operaciones aritméticas en sumas, descomponiendo las operaciones complejas en pasos elementales.

Actualmente la Unidad Aritmética ha cobrado gran importancia y el tipo de operaciones que realiza es muy complejo, siendo similar a una **calculadora científica de alta velocidad y gran precisión**.

Las operaciones que es capaz de realizar la ALU se clasifican en tres grupos:

Aritméticas. Normalmente son la suma (ADD), la resta (SUB), la multiplicación (MUL), la división (DIV) y el cambio de signo.

Lógicas. Normalmente son la negación (NOT), la suma lógica (OR), el producto lógico (AND) y la suma exclusiva (XOR).

Desplazamiento. Consiste en desplazar los bits de una palabra un determinado número de posiciones hacia la derecha o hacia la izquierda.

El número de operadores que puede aceptar la unidad aritmético-lógica también es variable, clasificándose en operadores **monádicos, diádicos y triádicos** si es capaz de trabajar con uno, dos o tres datos de forma simultánea respectivamente.

3.3. Unidades Funcionales de las CPU Actuales

3.3.1. Memoria Caché

Es fundamental en la arquitectura de cualquier sistema informático, ya que mejora el rendimiento almacenando los datos usados con mucha frecuencia ahorrando llamadas a la memoria RAM con sus consiguientes esperas:

La caché es mucho más rápida que la RAM: utiliza memoria Static RAM (SRAM) que no necesita refresco por estar formada íntegramente con biestables.

El bus que comunica RAM y microprocesador (FSB -Front Side Bus) es siempre más lento que la velocidad interna del microprocesador.

Se denominan con la letra L (de level) y un número desde el 1 en adelante a la memoria caché según si ésta se encuentra en el propio núcleo (L1), en el encapsulado, pero fuera de los núcleos (L2) o en una zona aún más periférica (L3, etc.) como por ejemplo la placa base.

3.3.2. Coprocesador matemático (FPU)

El microprocesador -desde el Intel 80486- incorpora una unidad para cálculos avanzados en coma flotante (FPU=Float Point Unit) relevando así a la ALU de dichos cálculos complejos.

3.3.3. Unidad de gestión de Memoria (MMU)

La MMU (Memory Management Unit) es la unidad encargada de traducir direcciones virtuales a direcciones físicas reales. Recordemos que los sistemas operativos trabajan actualmente con el modo protegido del procesador, que entre otras cosas ocultan direcciones reales de subrutinas y de memoria RAM.

3.3.4. Unidad de predicción de salto (Branch Predictor)

Un predictor de saltos es un circuito digital utilizado en los procesadores que utilizan **segmentación** de la unidad de proceso para reducir ciclos de parada en la segmentación.

Los saltos condicionales introducen retardo en estos procesadores, ya que normalmente no se evalúa la condición del salto hasta pasadas varias etapas, lo que hace que se tenga que parar el

cauce, o que se puedan introducir instrucciones en la segmentación que no deben de ser ejecutadas, teniendo que convertirse posteriormente en NOP, y decrementando así el rendimiento.

La predicción es posible anotando el comportamiento del programa en saltos anteriores.

Un predictor dinámico trabaja en tiempo de ejecución, intentando aprender el comportamiento del programa para predecir con la mínima tasa de fallos si un salto será o no tomado. Existen varios tipos dependiendo de la información que son capaces de recoger sobre el programa para hacer predicciones.

3.3.5. Unidad de instrucciones extendidas

Los procesadores -desde el Intel Pentium MMX- incorporan instrucciones complejas como las SSE2, para la gestión avanzada de multimedia y que hacen uso directo de esta unidad.

Últimamente están empezando a incorporar también instrucciones RISC para protección y seguridad (por ejemplo, para generar tokens en las tecnologías VPro de Intel).

3.3.6. Núcleos especializados y SoC

Otra tendencia de todos los microprocesadores es a transformarse en lo que se conoce como SoC, y que se emplea ampliamente en la tecnología ARM: Systems on a Chip.

De esta forma incorporan GPU Completas especializadas en gráficos (por ejemplo, Intel UHD o Nvidia GeForce) así como CPU especializadas en IA (núcleos neuromórficos)

No obstante, los microprocesadores basados en ARM van un paso más allá y pueden incorporar controladores bluetooth, wifi, sonido, etc.

3.4. Arquitectura del Sistema de Entrada/Salida

El conjunto de componentes de un ordenador se comunica entre sí mediante ciertos caminos, que constituyen la estructura de interconexión, o buses de Entrada/Salida

En la arquitectura Von Neumann se distinguen los buses de **datos, dirección y control**:

3.4.1. Bus de datos:

El bus de datos permite el intercambio de datos entre la CPU y el resto de unidades.

3.4.2. Bus de direcciones:

La memoria RAM es direccionable, de forma que cada celda de memoria tiene su propia dirección. Las direcciones son un número que selecciona una celda de memoria dentro de la memoria principal o en el espacio de direcciones de la unidad de entrada/salida.

El bus de direcciones es un canal del microprocesador totalmente independiente del bus de datos donde se establece la dirección de memoria del dato en tránsito.

El bus de dirección consiste en el conjunto de líneas eléctricas necesarias para establecer una dirección. La capacidad de la memoria que se puede direccionar depende de la cantidad de bits que conforman el bus de direcciones, siendo 2^n el tamaño máximo en bits del banco de memoria que se podrá direccionar con n líneas. Por ejemplo, para direccionar una memoria de 256 bits, son necesarias al menos 8 líneas, pues $2^8 = 256$. Adicionalmente pueden ser necesarias líneas de control para señalar cuándo la dirección está disponible en el bus. Esto depende del diseño del propio bus.

3.4.3. Bus de control:

El bus de control gobierna el uso y acceso a las líneas de datos y de direcciones. Como estas líneas están compartidas por todos los componentes, tiene que proveerse de determinados mecanismos que controlen su utilización. Las señales de control transmiten tanto órdenes como información de sincronización, evitando que haya colisión de información en el sistema.

3.4.4. Buses multiplexados:

Algunos diseños utilizan líneas eléctricas multiplexadas para el bus de direcciones y el bus de datos. Esto significa que un mismo conjunto de líneas eléctricas se comportan unas veces como bus de direcciones y otras veces como bus de datos, pero nunca al mismo tiempo. Una línea de control permite discernir cuál de las dos funciones está activa.

3.4.5. Periféricos.

Los periféricos se interconectan a la unidad de Entrada / Salida mediante el uso de una **interfaz** con especificaciones de acceso **estandarizadas** tanto a nivel de forma, como de voltaje y de protocolos de comunicación.

Cada periférico dispone de un **circuito controlador** (o controladora) que gestiona toda la lógica del mismo.

Las peculiaridades de cada dispositivo son gestionadas de forma transparente por el sistema operativo mediante un **software controlador (driver)** que es normalmente provisto por el propio fabricante.

Los periféricos pueden ser clasificados en dispositivos de **Entrada**, de **Salida**, o de Entrada y Salida; estos últimos son de tipo **almacenamiento** o **comunicación**.

4. Funcionamiento de la unidad central de proceso.

Un ciclo de instrucción (también llamado ciclo de fetch-and-execute) es el período que tarda la unidad central de proceso (CPU) en ejecutar una instrucción de lenguaje máquina.

Comprende una secuencia de acciones determinada que debe llevar a cabo la CPU para ejecutar cada instrucción en un programa. **Cada instrucción del juego de instrucciones de una CPU, puede requerir diferente número de ciclos de instrucción** para su ejecución. Un ciclo de instrucción está formado por uno o más ciclos máquina.

4.1. Fases de ejecución de las instrucciones.

1. Buscar la instrucción en la memoria principal (Fetch)

Se vuela el valor del registro **contador de programa** sobre el bus de direcciones. Entonces la CPU pasa la instrucción de la memoria principal a través del bus de datos al **Registro de Dirección de Memoria** (MAR). A continuación, el valor del MAR es colocado en el **Registro de Instrucción Actual** (CIR), un circuito que guarda la instrucción temporalmente de manera que pueda ser decodificada y ejecutada.

2. Decodificar la instrucción (Decode)

El **decodificador** de instrucción interpreta e implementa la instrucción. El registro de instrucción (IR) mantiene la instrucción en curso mientras el contador de programa (PC, program counter) guarda la dirección de memoria de la siguiente instrucción a ser ejecutada.

3. Ejecutar la instrucción (Execute)

A partir del registro de instrucción, los datos que forman la instrucción son **decodificados** por la unidad de control. Ésta **interpreta la información como una secuencia de señales** de control que son enviadas a las unidades funcionales de la CPU para realizar la operación requerida por la instrucción.

4. Escritura de los resultados (writeback)

El resultado generado por la operación es **almacenado en la memoria principal** o enviado a un dispositivo de salida dependiendo de la instrucción.

5. Finalización (retirement)

Basándose en los resultados de la operación, **el contador de programa se incrementa** para apuntar a la siguiente instrucción **o se actualiza** con una dirección diferente donde la próxima instrucción será recogida.

En los procesadores superescalares se añaden además las fases de planificación

El ciclo de búsqueda

Los pasos 1 y 2 del ciclo de instrucción se conocen como ciclo de búsqueda. El ciclo de búsqueda procesa la instrucción a partir de la palabra de instrucción, que contiene el código de operación y el operando.

El ciclo de ejecución

Los pasos 3 y 4 del ciclo de instrucción se conocen como ciclo de ejecución. Estos pasos cambiarán con cada tipo de instrucción.

4.1.1. Juego de instrucciones.

Según el tipo de instrucciones que implementan, éstas se han dividido históricamente en:

CISC (Complex Instruction Set Computer): Contienen un conjunto de instrucciones complejas, de tamaño variable.

RISC (Reduced Instruction set computer): Poseen un conjunto de instrucciones reducido, y de tamaño fijo. Este tipo de instrucciones posibilitan la segmentación y el paralelismo.

Como ejemplo encontramos los procesadores que siguen la arquitectura ARM (Advanced RISC Machine) muy utilizados actualmente en dispositivos móviles.

Según el autor J. Glenn Brookshear, en su libro Introducción a la computación, las instrucciones a su vez, se clasifican en tres grupos:

Transferencia de datos

Aritmético/lógicas

Control

Adicionalmente, hay que indicar que existen juegos de instrucciones estandarizados, los cuales la CPU indican en sus especificaciones si los soportan, por ejemplo: x86-64, o AVX2.

5. Ampliación

5.1. Formato de las instrucciones.

Las instrucciones se representan acorde a un formato, el cual especifica el significado de cada uno de sus bits. Normalmente, pueden contener:

- Operación
- Dirección de los operandos
- Dirección del resultado

Dirección de la siguiente instrucción

Para simplificar su decodificación, estos bits se agrupan en campos: códigos de operación, operando 1 y Operando 2, por ejemplo: ADD R1 R2, indicaría que se sume R1 con R2 y el resultado se almacene en R1.

5.2. Modos de funcionamiento de la CPU

5.2.1. Aritmética de segmento

Los procesadores 8086 usaban 20 bits de dirección que les permitía acceder hasta 1 MB de memoria. Sus registros internos eran de sólo 16 bits, por lo que se desarrolló un mecanismo usando registros de segmento para poder acceder al megabyte de memoria completo. En lugar de suministrar los 4 bits faltantes para direccionar los 20 bits, este procesador desplazaba el contenido del registro de segmento 4 bits hacia la izquierda y lo suman a una dirección de memoria de 16 bits proveniente de registros índice y opcionalmente una constante, siendo el resultado la dirección efectiva.

Este modo de funcionamiento tan artificioso se intenta superar en los siguientes procesadores introduciendo el modo de funcionamiento protegido, que aísla al programador del direccionamiento real de la memoria entre otras cosas.

5.2.2. Modo Real

El modo real (también llamado modo de dirección real en los manuales de Intel) es un modo de operación del 8086 y posteriores CPUs compatibles de la arquitectura x86. El modo real está caracterizado por 20 bits de espacio de direcciones segmentado, acceso directo del software a las rutinas del BIOS y el hardware periférico, y no tiene conceptos de protección de memoria o multitarea a nivel de hardware. Todos los CPUs x86 de las series del 80286 y posteriores empiezan en modo real al encenderse el computador; los CPUs 80186 y anteriores tenían sólo un modo operacional, que era equivalente al modo real en chips posteriores.

5.2.3. Modo Protegido

El modo protegido es un modo operacional de los CPUs compatibles x86 de la serie 80286 y posteriores.

El modo protegido tiene un número de nuevas características diseñadas para mejorar la multitarea y la estabilidad del sistema, tales como la protección de memoria -incluyendo la eliminación de la aritmética de segmentos-, y soporte de hardware para memoria virtual como también la conmutación de tarea.

En el 80386 y procesadores de 32 bits posteriores se agregó un sistema de paginación que es parte del modo protegido.

La mayoría de los sistemas operativos x86 modernos se ejecutan en modo protegido.

5.3. CISC versus RISC

5.3.1. Tecnología CISC

La arquitectura CISC (Complex Instruction Set Computer, Ordenador de Conjunto de Instrucciones Complejas) se refiere a la conexión permanente del procesador con las instrucciones complejas, difíciles de crear a partir de las instrucciones de base.

La arquitectura CISC es especialmente popular en procesadores de tipo 80x86. Este tipo de arquitectura tiene un costo elevado a causa de las funciones avanzadas impresas en la sílice.

Las instrucciones son de **longitud diversa**, y a veces requieren **más de un ciclo de reloj**. Dado que los procesadores basados en la arquitectura CISC sólo pueden procesar una instrucción a la vez, el tiempo de procesamiento es una función del tamaño de la instrucción.

Como, por ejemplo:

Intel 8086, 8088, 80286, 80386, 80486.

Motorola 68000, 68010, 68020, 68030, 6840.

5.3.2. Tecnología RISC

Los procesadores con tecnología RISC (Reduced Instruction Set Computer, Ordenador de Conjunto de Instrucciones Reducidas) no poseen funciones avanzadas conectadas en forma permanente.

Es por eso que los programas deben **traducirse** en instrucciones sencillas, lo cual complica el desarrollo o hace necesaria la utilización de un procesador más potente. Este tipo de arquitectura tiene un costo de producción reducido si se lo compara con los procesadores CISC. Además, las instrucciones de naturaleza sencilla se ejecutan **en un sólo ciclo de reloj**, lo cual acelera la ejecución del programa si se lo compara con los procesadores CISC. Para terminar, dichos procesadores pueden manejar múltiples instrucciones en forma simultánea, procesándolas en paralelo. Las ventajas de los procesadores RISC, especialmente las ligadas a los sistemas abiertos (UNIX), los hacen plataformas ideales para explorar los puntos fuertes de los **sistemas multiprocesadores**.

6. Conclusión

La arquitectura de los ordenadores actuales está basada en principios de los años cincuenta del pasado siglo XX; no por eso es algo inamovible. Actualmente la arquitectura CISC dominante en el sector durante muchos años bajo marcas como Intel, AMD, IBM, Cyrix y otros, ha tendido a una convergencia con un modelo híbrido con RISC, siempre bajo la arquitectura Von Neuman.

Sin embargo, cobra terreno la arquitectura RISC basada en Harvard, liderada por el fabricante de núcleos ARM, y no sólo en el ámbito de los smartphones y dispositivos Android y IOS; Apple ya ha anunciado (año 2020) que comenzará a fabricar ordenadores portátiles con procesadores de su propia marca basados en ARM en lugar de utilizar chips Intel.

A pesar de todo, vemos que los principios de la arquitectura del ordenador no varían mucho de las originales, siendo estas y el modo de funcionamiento algo que no ha cambiado apenas y quizás tengamos que esperar al desarrollo de ordenadores cuánticos a gran escala para ver ese cambio.

6.1. Relación del tema con el sistema educativo actual

Este tema es aplicado en el aula en los módulos profesionales siguientes, con las atribuciones docentes indicadas (PES/SAI):

Formación profesional básica

- Montaje y Mantenimiento de sistemas y componentes informáticos (TPB en Informática de Oficina/ TPB en informática y Comunicaciones) (PES/SAI)

Grado Medio

- Montaje y Mantenimiento de Equipos (SMR) (PES/SAI)

Grado Superior

- Sistemas informáticos (DAM / DAW) (PES/SAI)
- Fundamentos de hardware (ASIR) (PES/SAI)

ESO

- Tecnologías de la Información y la Comunicación I (Bachillerato) (PES)

7. Bibliografía

- *De Anasagasti, Miguel. "Fundamentos de la Computadora"* 9^aed 2004 Edt. Paraninfo
- *Patterson D.A. y Hennessy JL. "Estructura y diseño de computadoras: la interfaz hardware/software"* 4^a Ed. (2005) Edt McGraw-Hill
- *Prieto A, Lloris A, Torres JC. "Introducción a la Informática"* 4^aed. (2006) Edt. McGraw-Hill
- *Stallings W. "Organización y Arquitectura de Computadoras"* (2006) 5^a Ed. Edt. Prentice-Hall

- *Ramos A, Ramos MJ y Viñas S "Montaje y Mantenimiento de Equipos" (2012). Edt McGraw-Hill*
- *Jiménez Cembreras, Isabel Mª "Sistemas Informáticos" 2ªEd (2018) Edt. Garceta*
- *Moreno Pérez, JC. "Fundamentos del Hardware" (2019) Edt. Síntesis*
- *Gallego Cano JC y Otros. "Montaje y Mantenimiento de Equipos y Componentes Informáticos" 2018 Edt. Editex.*

