

TEMA 2

ELEMENTOS FUNCIONALES DE UN ORDENADOR DIGITAL

INDICE:

| | |
|--------------------------------------|---|
| 1. INTRODUCCIÓN | 1 |
| 2. MEMORIA PRINCIPAL | 1 |
| 2.1. Memoria ROM..... | 1 |
| 2.2. Memoria RAM..... | 2 |
| 3. UNIDAD CENTRAL DE PROCESO | 3 |
| 3.1. Ciclo de instrucción..... | 3 |
| 3.2. Unidad de control..... | 3 |
| 3.3. Unidad aritmético-lógica | 4 |
| 3.4. Otros componentes..... | 5 |
| 4. SISTEMA DE ENTRADA Y SALIDA | 6 |
| 4.1. Buses..... | 6 |
| 4.2. Interfaces | 7 |
| 4.3. Sincronización con la CPU..... | 7 |
| 5. CONCLUSIÓN | 7 |
| 6. BIBLIOGRAFÍA | 8 |
| 7. NORMATIVA..... | 8 |

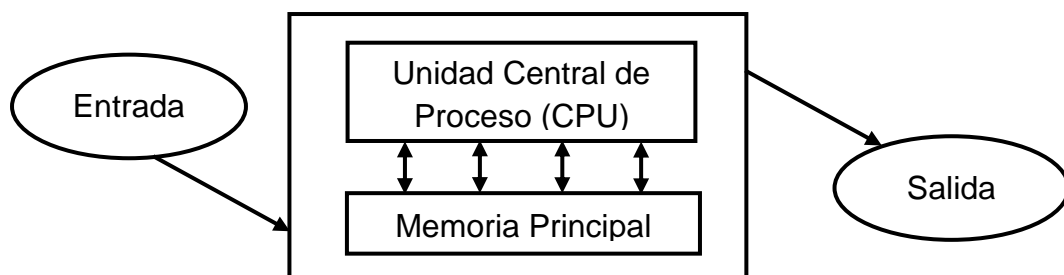
Realizado por Cayetano Borja Carrillo

Tiempo de escritura: 1 hora y 50 minutos

1. INTRODUCCIÓN

Los elementos funcionales o mínimos que debe tener un ordenador digital para poder funcionar están definidos en la arquitectura en la que se basa. Existen varias arquitecturas de computadoras, siendo la más extendida y la usada en la mayoría de los ordenadores que usamos a diario (PC, portátil, tableta, videoconsola, teléfono móvil, etc.) la arquitectura Von Neumann.

Dicha arquitectura establece que los elementos mínimos que debe tener un ordenador son: una memoria principal capaz de almacenar tanto datos como instrucciones, una unidad central de proceso o CPU que ejecute las instrucciones y un sistema de entrada y salida que permita la comunicación con el exterior.



En este tema se desarrollan el funcionamiento y las características de cada elemento funcional. Se trata de un tema de gran importancia dentro del campo de estudio del hardware, ya que son elementos fundamentales de un sistema informático.

2. MEMORIA PRINCIPAL

La memoria principal de un sistema informático es aquella memoria estrictamente necesaria para que un ordenador pueda funcionar y la única que es accedida directamente por la CPU, por lo que otras memorias como discos duros o pendrives no forman parte de la memoria principal.

Existen 2 tipos de memorias que, en conjunto, forman la memoria interna. Estas son la memoria ROM y la RAM.

2.1. Memoria ROM

Se llama memoria ROM (*Read Only Memory*) o memoria de sólo lectura a este tipo de memoria porque originalmente no permitía la escritura. Sin embargo, las ROM más modernas, como EPROM y Flash EEPROM se siguen llamando ROM, aunque se puedan escribir sobre ellas, porque es algo que no se hace con frecuencia.

La ROM consiste en una pequeña memoria con una serie de programas que se ejecutan al arrancar el ordenador. El *software* almacenado en la ROM se denomina *firmware* y se divide en 2 partes: la rutina POST y la rutina de arranque.

- Rutina POST: La rutina POST (*Power-On Self-Test*) o autoprueba de arranque se encarga de verificar el estado del *hardware* al encender el ordenador. Si se detecta algún error, el arranque se paraliza y emitirá una serie de pitidos o parpadeos de leds característicos para cada tipo de error. En caso de no detectar ningún problema, dará paso a la rutina de arranque.
- Rutina de Arranque (*Bootstrapping*): Esta rutina se encarga de buscar el núcleo del sistema operativo u otro *software bootable* en una unidad de almacenamiento secundario, como un disco duro, y lo carga en la RAM. A partir de este momento, el control del sistema es cedido al *software* cargado.

Desde la década de los 70, el *firmware* almacenado en esta memoria ha sido la BIOS (*Basic Input/Output System*) o sistema básico de entrada y salida, proporcionando una interfaz básica entre el usuario y el *hardware*.

A partir del 2010, aparece un nuevo estándar llamado UEFI (*Unified Extensible Firmware Interface*) como sustituto de la BIOS. UEFI realiza las mismas tareas que BIOS, pero además proporciona otras funciones como permitir el acceso remoto, soporte de discos con esquema de particiones GPT y, por tanto, soporte de particiones de más de 2TB, mayor velocidad de arranque, conectarse a Internet para actualizarse, permitir el manejo del ratón en su interfaz y mayor seguridad en el inicio gracias a “*Secure Boot*”, que impide la ejecución de aplicaciones no firmadas en el arranque del sistema.

2.2. Memoria RAM

La memoria RAM (*Random Access Memory*), o memoria de acceso aleatorio, es una memoria que almacena los datos e instrucciones que se están procesando o que ya han sido procesados. En términos generales, almacena el sistema operativo y los programas en ejecución. Es volátil, por lo que pierde toda la información cuando deja de recibir electricidad.

La RAM se organiza en una matriz de celdas capaces de almacenar 1 bit (un cero o un uno). Como un solo bit no da suficiente información, se agrupan varias celdas en la misma dirección formando una palabra de memoria. Por ejemplo, una memoria RAM con un ancho de palabra de 8 bits, tendrá 8 celdas por cada dirección de memoria.

La RAM se denomina “de acceso aleatorio” porque el tiempo que tarda en realizar una operación de lectura o escritura de una palabra, es siempre el mismo sin importar la dirección donde se encuentra, no siendo obligatorio seguir un orden secuencial para acceder a la información como si ocurre en algunas memorias secundarias.

El esquema de la memoria RAM es el siguiente:

| Dirección | Información | | | | | | | | |
|-----------|-------------|---|---|---|---|---|---|---|-------------------|
| 0000 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | Celda |
| 0001 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | |
| 0002 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | Palabra de 8 bits |
| ... | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | |

3. UNIDAD CENTRAL DE PROCESO

La unidad central de proceso o CPU (*Central Processing Unit*), es el circuito integrado más complejo de un sistema informático. Su función consiste en ejecutar instrucciones programadas en lenguaje de bajo nivel, realizando operaciones aritméticas y lógicas simples y accesos a memoria. En otras palabras, la CPU se encarga de ejecutar las instrucciones que recibe de la memoria RAM o de otros dispositivos como los periféricos.

De acuerdo a la arquitectura Von Neumann, los elementos mínimos que debe de tener una CPU son una unidad de control y una unidad aritmético-lógica. Antes de ver el funcionamiento de estos 2 componentes, veamos en qué consiste un ciclo de instrucción para explicarlo mejor.

3.1. Ciclo de instrucción

Un ciclo de instrucción es el periodo por el que pasa una instrucción durante su ejecución. Se compone de una serie de etapas siendo las más básicas las siguientes:

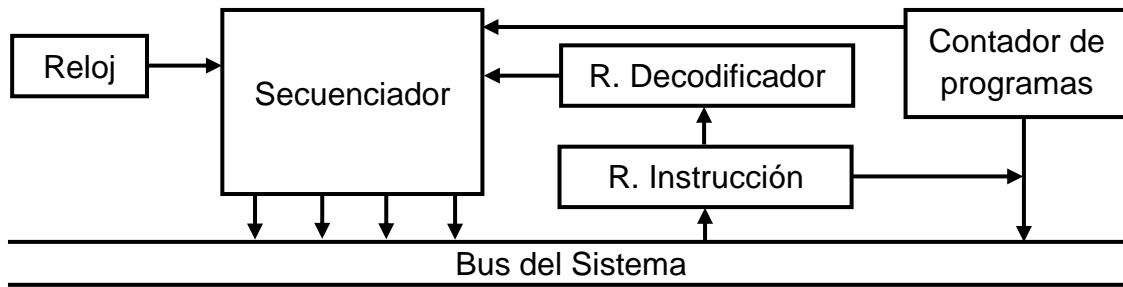
- Fetch (Lectura): La instrucción llega a la CPU y se coloca en los registros internos.
- Decode (Decodificación): La instrucción se descodifica y se interpreta su tipo (aritmética, lógica, de E/S, de control, de transferencia de datos, etc.).
- Execute (Ejecución): La instrucción se ejecuta.
- WriteBack (Guardado): Se almacena el resultado generado.

La ejecución de estas etapas permite paralelismo por segmentación. Esto quiere decir que se pueden ejecutar varias instrucciones a la vez, de forma que, mientras una instrucción está en la etapa "*Decode*", otra instrucción puede estar en la "*Fetch*".

3.2. Unidad de control

La unidad de control o UC se encarga de dirigir y coordinar la mayoría de las operaciones que se ejecutan. Su función es la de interpretar instrucciones, controlar el flujo de datos y garantizar que las operaciones se ejecuten en el orden correcto.

La estructura de una UC es la siguiente:



Su funcionamiento es el siguiente:

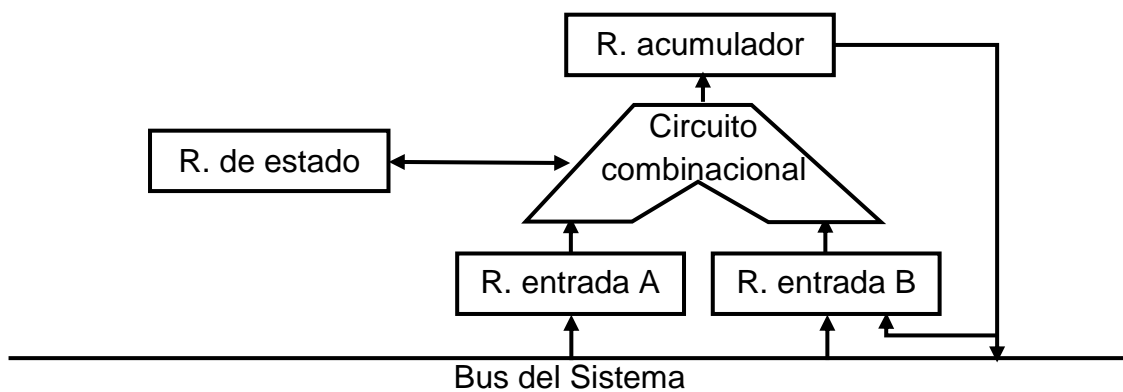
1. La instrucción a ejecutar viaja de la memoria RAM o de un periférico de entrada, según el tipo de instrucción, por el bus del sistema y se almacena en el registro instrucción (etapa *Fetch*).
2. La instrucción llega al registro decodificador donde se decodifica y se interpreta su tipo (etapa *Decode*). De forma simultánea, el contador de programas se actualiza con la dirección de la siguiente instrucción a ejecutar.
3. La instrucción llega al secuenciador, que genera microórdenes que permiten su ejecución (etapa *Execute*). Si la instrucción es de tipo lógico o aritmético, se hace uso de la ALU.
4. El resultado obtenido viaja del secuenciador a la memoria RAM o al periférico de salida, según la instrucción (etapa *Writeback*).

Todos estos pasos van dirigidos por impulsos eléctricos a intervalos constantes generados por el reloj. La frecuencia de estos intervalos se mide en hercios (Hz).

3.3. Unidad aritmético-lógica

La unidad aritmético-lógica o ALU es un dispositivo encargado de ejecutar las instrucciones aritméticas (suma, restas, etc.) y lógicas (AND, OR, NOT, XOR, etc.) que recibe de la UC. Para realizar la operación, la ALU necesita la orden indicada por la instrucción, la dirección donde están ubicados los datos u operandos y la dirección donde se almacenará el resultado.

La estructura de una ALU es la siguiente:



La función de cada elemento es la siguiente:

- Registro entrada A: Almacena el primer operando.
- Registro entrada B: Almacena el segundo operando.
- Registro de estado: Almacena condiciones que se quedaron pendientes en la última operación, como el acarreo.
- Registro acumulador: Almacena los resultados que se están llevando a cabo.
- Circuito operacional: Realiza la operación con los datos que hay en los registros.

3.4. Otros componentes

Además de la UC y la ALU, las CPU actuales incluyen otros componentes adicionales que mejoran el rendimiento. Estos componentes son los siguientes:

Unidad coma flotante

La unidad coma flotante o FPU (*Floating Point Unit*) es un dispositivo especializado en ejecutar operaciones aritméticas en coma flotante, es decir, con números reales.

Una ALU por si sola también es capaz de realizar estos cálculos, pero el coste computacional es muy alto, ya que tiene que emular la función mediante microcódigos, por lo que la FPU libera a la ALU de realizar este trabajo.

Al principio, las FPU eran coprocesadores separados de las CPU, pero a finales de los años 80, los fabricantes comenzaron a incorporar las FPU en el interior de las CPU. La primera CPU en tener una FPU integrada fue el Intel 80486DX.

Memoria caché

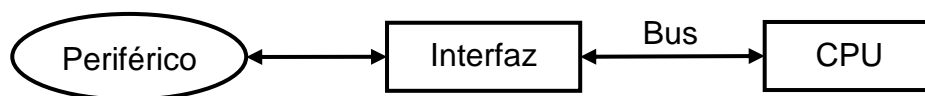
La caché es un conjunto de memorias (niveles) de gran velocidad, pero de poca capacidad, que hace de intermediaria entre la RAM y la CPU. Su función es la de almacenar temporalmente los datos que se prevé que van a ser consultados pronto o con más frecuencia. Los niveles de una memoria de caché pueden ser:

- Caché L1: Es la más pequeña y rápida de todas. Se compone de 2 memorias, una para almacenar exclusivamente datos y otra para instrucciones.
- Caché L2: De mayor capacidad que la L1, pero más lenta.
- Caché L3: Es más grande y lenta que las anteriores, pero aun así es muy veloz. Normalmente es compartida por todos los núcleos.
- Caché L4: Aunque no es habitual que un sistema incorpore esta memoria, algunos microprocesadores como el Intel i7-5775C la incluye para servir de apoyo a las gráficas integradas o iGPU.

Es importante tener en cuenta que no todos los microprocesadores incorporan en su interior los mismos niveles de caché. Por ejemplo, algunas CPU antiguas como el 80486 y el primer Pentium solo tienen una caché L1 integrada en el chip, aunque pueden utilizar de forma opcional una caché L2 ubicada en la placa base. En la actualidad, lo normal es que incorporen los 3 primeros niveles (L1, L2 y L3).

4. SISTEMA DE ENTRADA Y SALIDA

El sistema de E/S se trata de un mecanismo que permite a la CPU comunicarse con el exterior, es decir, con los periféricos. La conexión entre estos 2 elementos no es directa, sino que el periférico se conecta a una interfaz que comunica, a través de un bus, con la CPU.



4.1. Buses

Un bus de comunicaciones es un canal que permite la transferencia de datos entre dispositivos informáticos. Puede ser una línea metálica soldada en una placa de silicio, un cable, un conjunto de pines, resistores, condensadores, etc.

Dependiendo del tipo de transferencia, un bus puede ser de 2 tipos:

- Bus en serie: La información se transmite bit a bit. Algunos ejemplos son USB, PCI-Express y Serial ATA.
- Bus en paralelo: Permite transmisión de varios bits a la vez. Esto es posible porque están compuestos por varias líneas o canales de comunicación, de forma que, un bus de 8 canales puede transmitir 8 bits simultáneamente. Algunos ejemplos son ISA, PCI y ATA.

Independientemente de si se trata de un bus en serie o en paralelo, un bus está compuesto por 3 líneas también denominadas buses y son:

- Bus de datos: Transporta la información entre la CPU y el resto de los componentes.
- Bus de direcciones: Contiene la dirección de memoria RAM donde se encuentra la instrucción a ejecutar o la posición del periférico al que se desea acceder.
- Bus de control: Como un mismo bus puede ser compartido por varios dispositivos, existe la posibilidad de que diferentes flujos de datos que provienen de distintos componentes choquen y se mezclen. El bus de control hace de arbitraje y evita la colisión de datos.

4.2. Interfaces

Una interfaz permite la conexión de un periférico o dispositivo con el bus que comunica con la CPU. Existen 3 tipos de interfaces:

- Puerto: Interfaz externa que permite una conexión física del periférico. Algunos puertos son PS/2, USB, HDMI, VGA, DVI, Firewire y DisplayPort.
- Ranura de expansión: Interfaz interna donde el dispositivo se conecta a una ranura de expansión que hay en placa base. Estos dispositivos son tarjetas gráficas, de sonido, de red, etc. Algunas ranuras son ISA, PCI, AGP y PCI-E.
- Interfaz inalámbrica: El periférico se comunica con el ordenador a través de ondas electromagnéticas que viajan por el aire. El ordenador debe de tener un receptor que sea capaz de captar e interpretar dichas ondas. Algunos medios inalámbricos son Wi-Fi, Bluetooth, infrarrojos y NFC.

4.3. Sincronización con la CPU

La CPU tiene que estar permanentemente sincronizada con el sistema de entrada y salida para poder enviar o recibir información de los periféricos. Existen varias técnicas de sincronización como las siguientes:

- Sondeo o Polling: La CPU se encarga de sondear todos los periféricos cada cierto tiempo para comprobar si alguno tiene una petición. Este método es ineficiente, ya que se consume tiempo constantemente en realizar todas las instrucciones de sondeo. Los periféricos con interfaz USB 2.0 y versiones anteriores utilizan esta técnica de sincronización.
- Interrupciones: La CPU recibe una señal llamada IRQ (*Interrupt Request*) indicando que debe interrumpir el flujo de ejecución normal para pasar a ejecutar las peticiones del periférico. De esta forma, no se pierde el tiempo sondeando constantemente a los periféricos. Los periféricos con interfaz PS/2 o USB 3.0 y versiones posteriores usan este método.

5. CONCLUSIÓN

Este tema es un tema introductorio a la estructura de computadores, donde se han repasado los elementos funcionales de un ordenador y la relación que hay entre ellos.

En el transcurso del tiempo, los equipos informáticos se han ido volviendo más complejos y potentes, pero su estructura básica ha permanecido inalterada, ya que todos se basan en la misma arquitectura y están diseñados bajo los mismos principios teóricos. Quizás el próximo gran salto se produzca una vez se desarrollen los ordenadores cuánticos a gran escala.

6. BIBLIOGRAFÍA

- Mano, M. M. (1994). *Arquitectura de computadoras (3ª ed.)*. Pearson Educación.
- López Ureña, L. A. et al. (1997). *Fundamentos de Informática (1ª ed.)*. Rama.
- Prieto Espinosa, A. et al. (2006). *Introducción a la informática (4ª ed.)*. McGraw-Hill.
- Brookshear, J. G. (2012). *Introducción a la computación (11ª ed.)*. Pearson Educación.

7. NORMATIVA

Para el desarrollo de este tema, se ha tenido en cuenta la siguiente normativa, donde se especifican los contenidos, competencias y criterios de evaluación de los Ciclos Formativos y Bachillerato en Andalucía:

- Orden 7 de julio de 2009 (SMR). La parte correspondiente al módulo “Montaje y Mantenimiento de Equipos”.
- Orden 19 de julio de 2010 (ASIR). La parte correspondiente al módulo “Fundamentos del Hardware”.
- Orden 16 de junio de 2011 (DAW/DAM). La parte correspondiente al módulo “Sistemas Informáticos”.
- Instrucción 13/2022 (Bachillerato). La parte correspondiente a la asignatura “Tecnologías de la Información y Comunicación”