

## UNIDAD 2.

# Elementos funcionales de UN ORDENADOR

Los sistemas informáticos  
CFGS DAW

Sergio García / Alfredo Oltra

[Sergio.garcia@ceedcv.es](mailto:Sergio.garcia@ceedcv.es)

[alfredo.oltra@ceedcv.es](mailto:alfredo.oltra@ceedcv.es)

2019/2020

Versión: 190917.0943

## Licencia



**Reconocimiento - NoComercial - Compartirlgual (by-nc-sa):** No permite en sí ONU USO comercial de la obra original, ni de las obras Posibles

Derivadas, La Distribución de las Cuales se Dèbe Hacer con licencia Una Igual a La que regula la Obra originales.

## nomenclatura

A lo largo de Este tema se utilizarán Distintos Símbolos para distinguir Elementos Importantes Dentro del contenido. Símbolos Estós hijo:

- Importante

- Atención

- interesante

## ÍNDICE

<b>1. Evolución histórica .....</b>	<b>4</b>
1.1 ordenadores no digitales .....	4
1.2 Los ordenadores digitales .....	4
1.2.1 ordenadores de propósito general .....	5
1.2.2 transistores y circuitos integrados .....	5
<b>2. Definición del sistema informático .....</b>	<b>5</b>
<b>3. arquitecturas de computadora .....</b>	<b>6</b>
3.1 Arquitectura de von Neumann .....	6
3.2 Arquitectura de Harvard .....	7
<b>4. Los elementos funcionales de un ordenador .....</b>	<b>8</b>
4.1 CPU (Unidad Central de Proceso) .....	8
4.2 La unidad de memoria .....	9
4.3 Otros tipos de memoria .....	9
4.4 I / O (/ dispositivos externos de salida de entrada) .....	10
4.5 Buses .....	10
4.6 conjunto de instrucciones .....	11
<b>5. ciclo de instrucción .....</b>	<b>11</b>
<b>6. Ejemplo de un conjunto de instrucciones .....</b>	<b>12</b>
<b>7. Diseño de la CPU: RISC y CISC .....</b>	<b>13</b>
7.1 Los equipos RISC .....	14
7.2 ordenadores CISC .....	14
7.3 RISC vs CISC .....	14
<b>8. El material adicional .....</b>	<b>15</b>
<b>9. Bibliografía .....</b>	<b>15</b>

## UD02. Elementos funcionales de UN ORDENADOR

### 1. EVOLUCIÓN HISTÓRICA

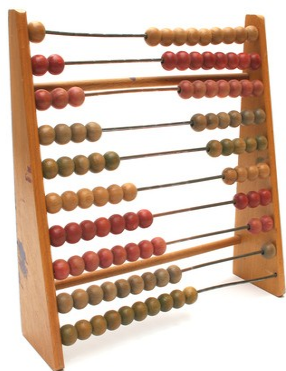


Figura 1. Abacus

#### 1.1 ordenadores no digitales

El ábaco fue inventado en el año 500 aC y fue el primer ordenador de la historia. Fue creado para realizar operaciones aritméticas rápida y era la máquina calculadora más rápida desde la llegada de otras calculadoras mecánicas, como Pascaline creado por Blaise Pascal (siglo XVII) o Bagage máquina creada por Charles Bagage (siglo XIX). Estos eran máquinas con elementos mecánicos que realizan operaciones aritméticas básicas.

Además, se establecieron los fundamentos de las computadoras modernas: código binario (Gottfried Leibniz, siglo XVII),

Bool álgebra (George Boole, siglo XIX) y la máquina de Turing (Alan Turing, 1936).

- Bool define las normas para trabajar en un sistema algebraico binario, es decir, en la que hay sólo dos dígitos (0 y 1) y no 9 como en el sistema decimal. De esta manera cualquier cantidad sólo puede ser representado mediante la definición de dos estados. Este sistema es básico en los ordenadores de hoy en día, que contienen elementos que se pueden encontrar en estos dos estados <sup>1</sup>.

#### 1.2 Los ordenadores digitales

La energía eléctrica nos dio una nueva manera de construir ordenadores más fiable y potente.

El primer ordenador programable binaria era Z1, creado por el ingeniero alemán Konrad Suze en 1938. En EE.UU., John Atanasoff y Clifford Berry creados en ABC 1939. Estas fueron las primeras máquinas que utilizan interruptores eléctricos para almacenar la información (un “botón de apagado” significa cero, un “interruptor en” uno entiende). Más tarde, había varios ordenadores programables como Colossus (1943) y ENIAC (1946) que fueron los primeros en utilizar tubos de vacío <sup>2</sup> para almacenar información. El problema de esas computadoras es que eran difíciles de re-programa, porque la programación se llevó a cabo la modificación de hardware con la mano y era difícil y costoso.

<sup>1</sup> Vamos a estudiar el sistema binario en la unidad 2

<sup>2</sup> tubos de vacío son dispositivos que controla la corriente eléctrica entre los electrodos en un recipiente evacuado. Acostumbrado a representar 0 y 1.

### 1.2.1 ordenadores de propósito general

En 1946 se creó EDSAC. Fue la primera máquina con instrucciones internas, haciendo fácil ser re-programada con la mano.

En 1949 EDVAC nació y fue el primer ordenador con programas almacenados. Fue una revolución porque ese equipo era capaz de ejecutar diferentes tipos de programas sin modificar su hardware. Fue el nacimiento de "software".

### 1.2.2 circuitos integrados y transistores

Los tubos de vacío tenían varios problemas. Eran grandes, poco fiables y consumen mucha energía, por lo que los ingenieros tenían que encontrar un sistema para reducirlos. En 1947 el transistor fue inventado y que reemplazó los tubos de vacío, reduciendo el tamaño de las computadoras, el ahorro de energía y hacer que los transistores más fiables fueron un éxito pero, al igual que los tubos de vacío, tienen que ser cableado para conectar todos los componentes juntos. Para resolver ese problema se inventaron los circuitos integrados (IC) en 1949. Ellos pusieron varios transistores en un solo paquete. Transistores y circuitos integrados ayudan a reducir los ordenadores durante la década de 1960.

En 1970 nacieron los ordenadores personales. Había un montón de ellos (MITS Altair, Apple II, IBM 5100, Marcos 8, ...) pero el mayor éxito fue IBM PC (1981). Fue la base de la mayoría de los ordenadores personales que utilizamos hoy en día. En 1990, los ordenadores personales están empezando a estar conectado a Internet. En 2000 la mayoría de ellos tienen acceso a Internet.

Hoy en día, hay ordenadores de todo el mundo (PC de escritorio, PC portátil, teléfonos móviles, tablets, relojes inteligentes, dispositivos de acondicionamiento físico, Smart TV ...).

## 2. DEFINICIÓN DE SISTEMA DE ORDENADOR

¿Qué es un sistema informático? El hombre siempre ha querido ordenadores de desarrollo que ayuda a resolver sus problemas. Pero necesitaban para construir una máquina para cada problema.

La idea era: ¿se puede construir una máquina para resolver diferentes problemas? La respuesta es: sí, podemos. Esta máquina se llama un sistema informático y su principal característica es que se compone de dos partes: el hardware (los elementos físicos) y el software (las instrucciones que le indican al hardware de lo que tiene que hacer, cómo resolver el problema). A veces, algunos autores incluyen dos piezas más: el usuario y los datos.

En esta unidad didáctica vamos a hablar acerca de los elementos funcionales de un ordenador, es decir, sobre los elementos conceptuales que forman parte de cualquier sistema informático y que en la vida real será parte del hardware.

### 3. arquitecturas de computadora

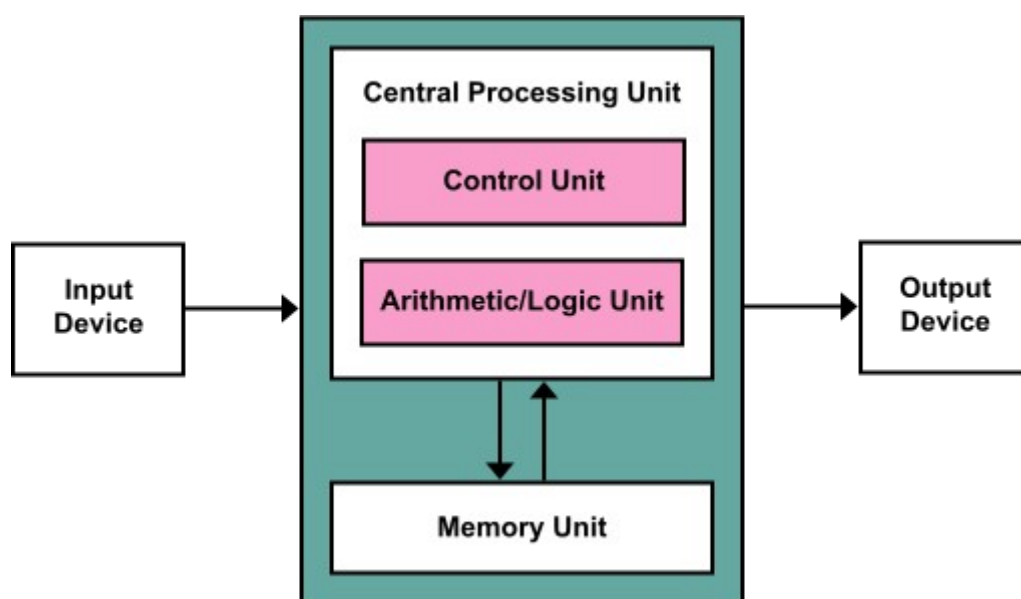
#### Arquitectura 3.1 von Neumann

Arquitectura de von Neumann es un modelo abstracto creado en 1946 que describe una arquitectura de ordenador. En la arquitectura Von Neumann los principales elementos de un ordenador son:

- CPU (Unidad Central de Proceso)
- Unidad de memoria
- Los autobuses
- I / O (entrada / salida a dispositivos externos) Arquitectura

obras Von Neumann como se describe:

- Los componentes están conectados por buses.
- Se puede ejecutar instrucciones elementales (código máquina).
- Las instrucciones se almacenan en la unidad de memoria. También los datos tienen que ser almacenados en la unidad de memoria.
- Ambas instrucciones y los datos podrían ser proporcionados por los dispositivos de entrada o generados por otros programas (compilador, enlazador, ...).
- La CPU lee una instrucción y la unidad de control genera cada señal para llevar a cabo la instrucción.
- Si es necesario, la CPU puede realizar operaciones de aritmética y lógica.
- Los resultados de la CPU pueden ser almacenados en la unidad de memoria o que pueden ser enviados a los dispositivos de salida.



arquitectura Figura 2. Von Neumann

- Hoy en día, hay varias mejoras de esta arquitectura, incluyendo características tales como E / S mapeada (unidad de memoria y dispositivos E / S son tratados como el mismo de la memoria).

### 3.2 Arquitectura de Harvard

arquitectura Harvard es otro modelo abstracto que describe una arquitectura de computadores. Se implementó menos que la arquitectura Von Neumann. Las principales diferencias entre la arquitectura Harvard y Von Neumann arquitectura son:

- En la arquitectura Von Neumann, los datos y las instrucciones se almacenan en la misma memoria. CPU sólo puede leer una instrucción o un conjunto de datos al mismo tiempo. En la arquitectura de Harvard, los datos y las instrucciones se almacenan por separado para cada una tiene su propio bus. CPU puede leer ambos al mismo tiempo (una instrucción y un conjunto de datos).
- En la arquitectura Von Neumann, los datos y las instrucciones se almacenan en la misma memoria y comparten el espacio de direcciones. En la arquitectura de Harvard, que se almacenan en memorias diferentes.

- Generalmente, ordenador de verdad no usan pura Von Neumann la arquitectura o arquitectura Harvard puro. La única son modelos abstractos.

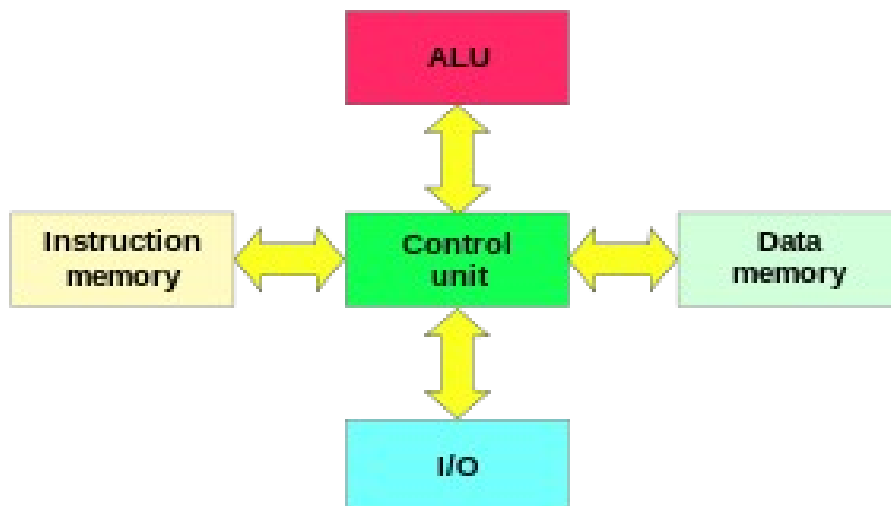


diagrama de la arquitectura Figura 3. Harvard

## 4. elementos funcionales de UN ORDENADOR

En este punto, vamos a describir los elementos funcionales de un ordenador que implementa la arquitectura Von Neumann.

### 4.1 CPU (Unidad Central de Proceso)

La CPU es el núcleo de la computadora. CPU lee las instrucciones de la memoria y ejecutar ellos, haciendo las operaciones requeridas. Se compone de:

- **registros:** que son pequeños y rápidos recuerdos. Se utilizan como almacenamiento temporal cuando se está ejecutando una instrucción o para almacenar temporalmente los resultados de operaciones.

Los registros más importantes son:

- **contador de programa (PC):** un contador incremental que almacena la dirección de memoria de la siguiente instrucción que se va a ejecutar.
- **registro de direcciones de memoria (MAR):** almacena la dirección de un bloque de memoria para leer o escribir a.
- **Registro MDR (MDR):** un registro de dos vías que almacena los datos obtenidos de la memoria (y listo para la CPU a proceso) o los datos a la espera de ser almacenado en la memoria.
- **registro de instrucción (IR):** un registro temporal para almacenar la instrucción que acaba de ser obtenen de la memoria.
- **registros de propósito general:** estos registros se utilizan para almacenar información temporal, por ejemplo para almacenar operandos de y adición o para almacenar el resultado de una operación aritmética-lógica.
- **banderas:** Estos son registros muy simples. Sólo pueden estar en dos estados. Encendido o apagado. Se utilizan para indicar el estado de una operación. Por ejemplo, la bandera Z se establece en 1 si la operación se ha traducido en 0., N se establece en 1 si el resultado es negativo.
- **bus interno:** es un bus interno que los registros se conectan, unidad de control y unidad aritmética-lógica.
- **La unidad de control (UC):** se decide qué hacer en cada momento. Es el *jefe*.  
Sus componentes son:
  - **Descifrador:** su función es decodificar las instrucciones y preparar todas las señales para ejecutar el programa correctamente. Por ejemplo una instrucción puede ser ADD (lo que suma dos números que están en la memoria). Cuando se trata de la ejecución, el decodificador genera una gran cantidad de pequeñas señales, tales como leer el primer número de la memoria, llevándolo a la ULA, leyendo el segundo número, llevándolo a la ULA, perfom la operación



(Añadir) y guarde el resultado en la memoria. En este caso una instrucción ha generado seis señales<sup>3</sup>.

- **Reloj:** es el sistema que establece el ritmo con el que se hacen las cosas. Es como la batuta del director. I permite que todos los componentes del sistema para operar perfectamente juntos.
- **Unidad aritmética y lógica (ULA):** es una unidad diseñada para realizar operaciones aritméticas (suma, resta, multiplicación, ...) y operaciones lógicas (no, y, o, ...). Es el **trabajador**.

Para ejecutar una instrucción de la CPU sigue estos pasos:

- Utilizar el **Contador de programa** a buscar a la nueva instrucción de la memoria.
- Decodificar la instrucción y obtener los datos de la memoria si es necesario.
- Realizar la instrucción y almacenar los datos en un registro interno o la memoria. El ciclo de instrucción

se detallará en adelante en la sección **ciclo de instrucción**.

- Para medir la velocidad de una CPU usamos unidades de frecuencia que indican el número de veces que se repite una cosa por unidad de tiempo. En geneal usamos la Hz (o sus múltiplos) que indica el número de operaciones que se pueden realizar en un segundo.

#### 4.2 La unidad de memoria

La unidad de memoria es una memoria interna que contiene las instrucciones para la ejecución y los datos o programas que utilizan. Consta de varias células que almacenan 0 o 1 (1 bit).

Un conjunto de células de un tamaño determinado se llama **palabra**. Computers suelen utilizar palabras de 32 bits o 64 bits. Cada palabra tiene una dirección asociada con él. Cuando se lee la información de la unidad de memoria, usted tiene que leer toda la palabra seleccionada. Por ejemplo, una memoria de 192 bits tiene 6 palabras de 32 bits. Las direcciones de memoria de esos 6 palabras serán de 0 a 5. La primera palabra irá desde el bit 0 a 31, el segundo de 32 a 63, etc.

Por lo general, esta es una memoria RAM (memoria de acceso aleatorio). RAM es una memoria volátil (se necesita energía para almacenar información, sin la energía de la información se ha ido). También el tiempo que se tarda en leer / guardar una célula es el mismo para todas las palabras.

- velocidad de la memoria se mide por el número de palabras que se pueden leer / tienda en una unidad de tiempo. Por lo general se mide en Hz (número de palabras que se pueden leer / almacenada en un segundo).

#### 4.3 Otros tipos de memoria

Otros tipos de memoria presente en una computadora son:

<sup>3</sup> En este caso, el proceso de Es simplifica. En realidad se generan muchas más señales

- **ROM:** Memoria de sólo lectura. Es una memoria de sólo lectura. No se puede borrar. Por lo general, se utiliza en la vieja BIOS.
- **EPROM:** Programable y borrrable memoria de sólo lectura. Es una memoria no volátil. Es una memoria de sólo lectura, pero puede ser borrado (con luz ultravioleta) y re-escrito. Por lo general, se utiliza en el BIOS, que son comúnmente leer, pero rara vez escrito.
- **EEPROM:** Eléctricamente programable y borrrable memoria de sólo lectura. Es una memoria no volátil. Es una memoria de solo lectura, pero puede ser borrada eléctricamente y re-escrito. Por lo general, se utiliza en el BIOS ROM, que son comúnmente leer, pero rara vez escrito.
- **Destello:** es una evolución de la EEPROM. Le permite escribir y leer en varios lugares de la memoria al mismo tiempo. Por lo general, se utiliza en las memorias USB. Hoy en día es la tecnología más utilizada para almacenar equipo de BIOS.

#### 4.4 I / O (Input / dispositivos externos de salida)

Los dispositivos de E / S son los dispositivos utilizados para obtener información (dispositivos de entrada) o para proporcionar información (dispositivos de salida). Se llaman generalmente los dispositivos periféricos. Un ratón o un teclado son ejemplos de dispositivos de entrada y una impresora o una pantalla son ejemplos de dispositivos de salida. También hay dispositivos que son la entrada y salida al mismo tiempo (un disco duro, una pantalla táctil, componentes internos, como una tarjeta de sonido, tarjeta gráfica, puerto de expansión, ...).

- Hay un truco para saber si un dispositivo es un periférico o no. Si es parte de la máquina de Von Neumann no es un periférico. De lo contrario, lo es.

#### 4.5 Los autobuses

Un bus es un conjunto de cables que conectan los componentes del ordenador. Un bus se define por su límite (número de bits que pueden ser enviados en una sola operación). Por lo general, es de 32 bits, 64 bits, 128 bits, ....

- La velocidad del bus se puede medir en el número de operaciones que se pueden realizar en una unidad de tiempo. Por lo general, se mide en Hz (operaciones que se pueden realizar en un segundo).

Físicamente, los autobuses se pueden clasificar como:

- **bus de serie:** autobuses envían información por un hilo cada vez. Son muy simples y requieren un hardware fácil de gestionar. Por esta razón, a pesar de utilizar un solo cable, el obtener un rendimiento muy bueno. Por ejemplo, SATA y USB son buses serie.
- **bus paralelo:** autobuses envían información en paralelo, por varios cables. Son complejos de proceso y requieren hardware complicado. Por lo general, se obtiene

menor rendimiento que el bus serie. Por ejemplo, el antiguo puerto COM o el puerto de impresora de edad fueron buses paralelos.

Por el lugar donde los autobuses están colocados los autobuses se pueden clasificar como:

- **Interno:** conectar “componentes internos” de un componente. Por ejemplo, “componentes internos” de CPU: Registros, unidad aritmética-lógica y la unidad de control.
- **Externo:** componentes de conexión, por ejemplo CPU con unidad de memoria, CPU con dispositivos de E / S, etc.

Arquitectura de von Neumann distingue entre tres buses externos:

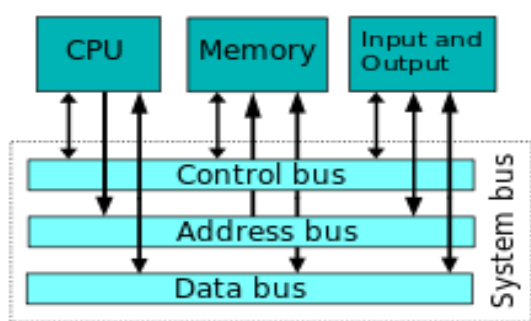


Figura 4. Buses

- **Bus de control:** que lleva las señales de control desde la unidad de control de la CPU.
- **bus de direcciones:** eso lleva el dirección de memoria para leer la información / tienda.
- **Bus de datos:** que transporta los datos y de instrucciones para ser leídos / almacenado.

#### set 4.6 Instrucción

CPU tiene un conjunto de instrucciones que se pueden realizar. Las instrucciones pueden ser:

- operaciones de lógica de cálculo: realizar operaciones aritméticas de lógica.
- Manejo de datos y la memoria operaciones: leer y almacenar la fecha de / a la memoria y desde / hacia los dispositivos de entrada / salida.
- las operaciones de control de flujo: las operaciones que cambian el flujo del programa de modificación de la **Contador de programa** (ORDENADOR PERSONAL).

### 5. ciclo de instrucción

Una instrucción es los datos que interpreta CPU para realizar una operación. Cuando hablamos de una CPU de 16 bits, 32 bits, 64 bits, ... estamos hablando de la cantidad de bits de una instrucción puede utilizar. Por lo general, una instrucción se divide en dos campos:

- código de operación: es el código que indica la CPU del tipo de operación a realizar.

- datos de la operación: son los datos asociado a la operación que se va a realizar. El ciclo de instrucción son los pasos que toma la CPU

para obtener el siguiente la instrucción y los cumplirán.

El ciclo de instrucción es similar en todos los modelos de ordenador. Los pasos son los siguientes:

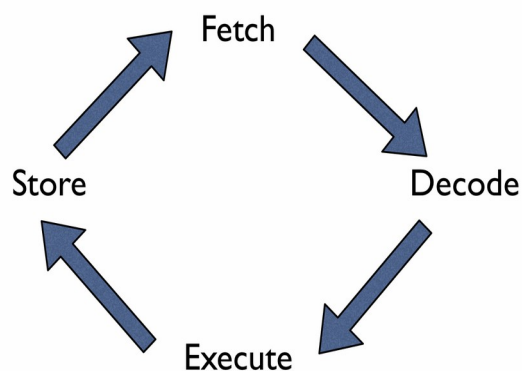


Figura Ciclo 5. Instrucción

- 1. Buscar la instrucción:** el siguiente instrucción se extrae de la dirección de memoria que es almacenado en el contador de programa (PC) y se almacena en la Instrucción Register (IR). Al final de esta operación, el contador de programa (PC) apunta a la siguiente instrucción (que se puede leer en la siguiente iteración del ciclo).
- 2. Decodificar la instrucción:** durante este ciclo de la instrucción codificada presente en la Instrucción Register (IR) se interpreta por el decodificador de la CPU.
- 3. Ejecutar la instrucción:** La unidad de control de la CPU pasa la información decodificada como una secuencia de señales de control para llevar a cabo las acciones requeridas por la instrucción. (Ejemplo: la lectura de los valores de los registros, pasarlos a la ALU, la escritura la parte posterior resultado a un registro).
- 4. Tienda:** El resultado generado por la operación se almacena en la memoria principal, en un registro o enviado a un dispositivo de salida.

El contador de programa (PC) puede ser actualizado a una dirección diferente de la que la siguiente instrucción será descabellada. (Es la manera de saltar de un lugar del código a otro). Cuando termina el ciclo, se repite de nuevo.

## 6. Ejemplo de un conjunto de instrucciones

En este punto vamos a describir un conjunto de ficción de instrucciones de un ordenador de ficción para entender lo que es un conjunto de instrucciones y cómo una instrucción está codificada.

En nuestro equipo de ficción:

- Cada instrucción está codificada utilizando 8 bits. Los primeros 2 bits se utilizan para almacenar la operación y los próximos 6 bits se utilizan para almacenar la información de la operación relacionada.
- Tenemos una memoria con  $2^6$  direcciones, cada una con una palabra de 8 bits (1 byte).
- Nuestra CPU tiene 3 registros temporalmente R1, R2 y R3. Pueden almacenar 8 bits.

- En nuestra memoria, la dirección 000000 contiene la palabra 00001001 (9 en complemento a dos) y 000001 contiene la palabra 00001000 (8 en complemento a dos).

En nuestro sistema de instrucción, hay 4 instrucciones:

- Instrucción 0: en binario 00. Se indica que podemos leer 2 números de registros de CPU temporalmente R1 y R2 y añadirlos. El resultado de la suma se almacena en R3. En esta instrucción sólo se usan 2 primeros bits, los próximos 6 bits se ignoran.
- Instrucción 1: en binario 01. Se utiliza una dirección de memoria. El contenido de esa dirección de memoria se almacena en R1.
- Instrucción 2: en binario 10. Se utiliza una dirección de memoria. El contenido de esa dirección de memoria se almacena en R2.
- Instrucción 3: en binario 11. Se muestra en la pantalla el valor de la R3 temporalmente registrarse. En esta instrucción sólo se usan 2 primeros bits, los próximos 6 bits se ignoran.

Por ejemplo, si queremos codificar una instrucción para la lectura de la palabra en la dirección de memoria 000010 y almacenarlo en R2, se utiliza la siguiente instrucción:

10 000010

Cuando los primeros 2 bits indican la operación (10) y los próximos 6 bits indican la dirección de memoria para leer.

El siguiente código mostrará en pantalla un número 17.

01 000000	Cargar en el contenido de R1 de dirección de 000.000.
10 000001	Cargar en el contenido de R2 de dirección de 000.001.
00 000000	Realizar la adición $R1 + R2$ y lo guarda en R3.
11 000000	Mostrar el resultado (R3) en la pantalla.

## 7. El diseño de la CPU: RISC y CISC

RISC y CISC son el método más popular para ordenadores de diseño. Se determina cómo es el conjunto de instrucciones de la CPU.

### 7.1 Los equipos RISC

**R**elucida **y**o **n**struction **S**et **C**omputer (RISC) es un diseño por ordenador con esas características:

- Conjunto reducido de instrucciones.
- Instrucciones de tamaño fijo.
- Sólo carga / almacenar instrucciones acceso a la memoria.

### 7.2 ordenadores CISC

**C**omplex **y**o **n**struction **S**et **C**omputer (RISC) es un diseño por ordenador con esas características:

- Alta conjunto de instrucciones.
- instrucciones de tamaño variable.
- instrucciones complejas, haciendo varias cosas en una sola instrucción o hacer operaciones complejas.

### 7.3 RISC vs CISC

RISC es una mejora a las computadoras CISC.

Por lo general, la principal ventaja del sistema operativo CPU RISC es que obtengan un mejor rendimiento que las CPU CISC (Se debe a que las CPU RISC son más simples que las CPU CISC y se puede obtener una mayor frecuencia de reloj).

Una ventaja del equipo CISC es que un programa requiere menos instrucciones para realizar la misma operación, ahorro de espacio y hacer más simple compiladores. CPU modernas que tienen un diseño CISC, en el interior tienen un diseño RISC y CISC las instrucciones más complejas están separados en varias instrucciones RISC. Por ejemplo, X86 CPU CPU CISC son en diseño, pero internamente son RISC CPU.

- ¿Por X86 no cambian de un diseño CISC a RISC un diseño? Porque si cambian el conjunto de instrucciones, que no sería compatible a nivel binario con las CPU anteriores de su familia.

## 8. material adicional

[1] Glosario. [2]

Ejercicios

## 9. BIBLIOGRAFÍA

[1] arquitectura Von Neumann

[https://en.wikipedia.org/wiki/Von\\_Neumann\\_architecture](https://en.wikipedia.org/wiki/Von_Neumann_architecture)

[2] arquitectura Harvard

[https://en.wikipedia.org/wiki/Harvard\\_architecture](https://en.wikipedia.org/wiki/Harvard_architecture)

[3] arquitectura Harvard Modificado

[https://en.wikipedia.org/wiki/Modified\\_Harvard\\_architecture](https://en.wikipedia.org/wiki/Modified_Harvard_architecture)

[4] ciclo de instrucción

[https://en.wikipedia.org/wiki/Instruction\\_cycle](https://en.wikipedia.org/wiki/Instruction_cycle)

[5] RISC

[https://en.wikipedia.org/wiki/Reduced\\_instruction\\_set\\_computing](https://en.wikipedia.org/wiki/Reduced_instruction_set_computing)

[6] CISC

[https://en.wikipedia.org/wiki/Complex\\_instruction\\_set\\_computing](https://en.wikipedia.org/wiki/Complex_instruction_set_computing)

[7] ordenador

<https://en.wikipedia.org/wiki/Computer>