

# Capítulo 1

## Introducción a los computadores

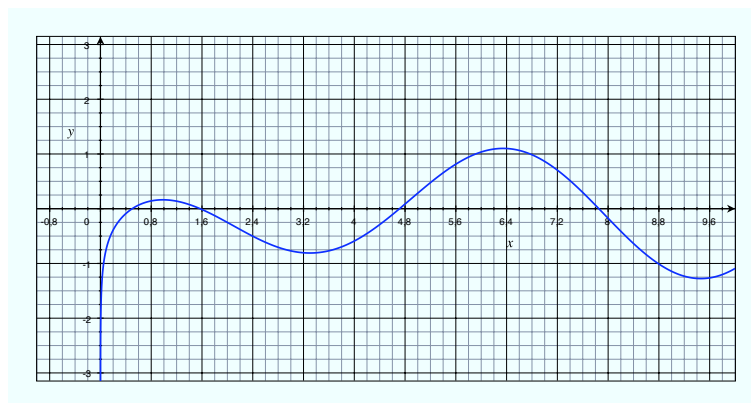
### 1.1. Introducción

#### 1.1.1. ¿Qué es un computador?

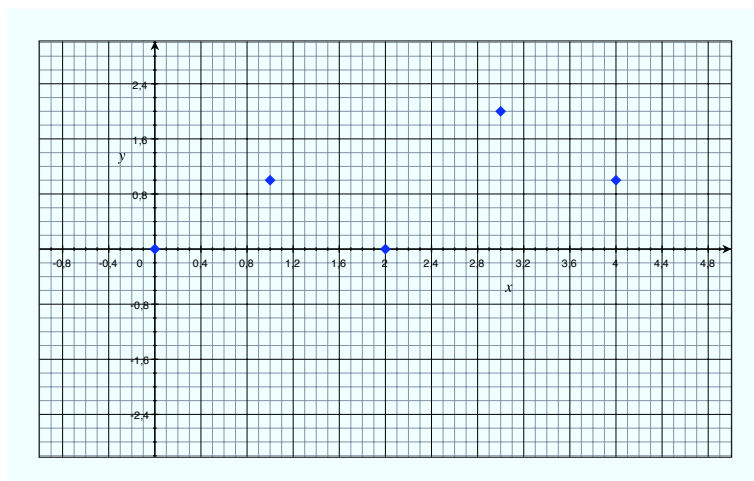
Un computador es un dispositivo físico programable (máquina) capaz de procesar información. Manipula y transforma la información para resolver un problema determinado. Hay una gran variedad de sistemas que pueden recibir correctamente el nombre de computador, desde los microprocesadores de un solo chip a los supercomputadores [4, 5, 3].

La información puede ser de dos tipos:

- **Analógica:** tiene un rango de variación continuo pudiendo tomar cualquier valor dentro de un rango definido (Fig. 1.1). Por ejemplo, la velocidad de un coche.
- **Digital:** sólo puede tomar valores discretos y finitos dentro de un rango (Fig. 1.2). En



**Figura 1.1:** *Ejemplo de información analógica.*



**Figura 1.2:** *Ejemplo de información digital.*

el ordenador, la información se guarda en forma de 0s y 1s. Todo son 0s y 1s: datos, instrucciones, señales de control, etc.

### 1.1.2. Estudio de su estructura

¿Por qué se estudia la estructura de un computador? Existe una preocupación de los programadores por el rendimiento de sus programas; intentan que sean rápidos. En dicho rendimiento influye la memoria, la forma en que se ejecutan los programas (el procesador), etc. Por ello tienen que tener conocimientos sobre la estructura del computador, sobre su funcionamiento interno, para así programar de acuerdo a sus características y lograr programas más rápidos [4, 5, 3].

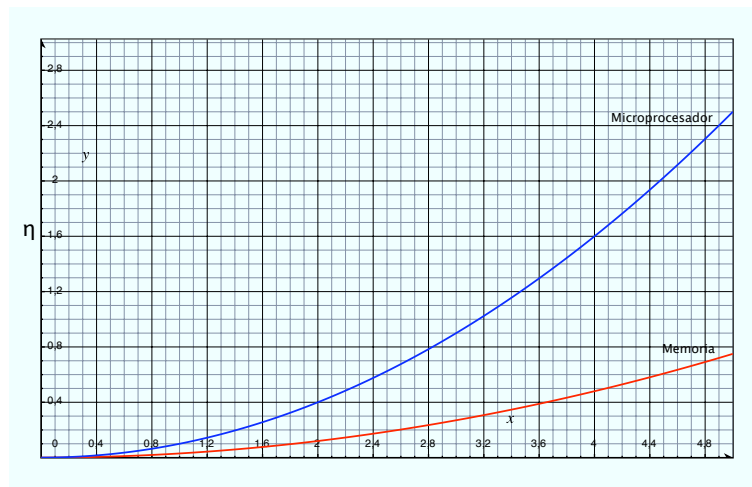
#### Evolución histórica

Desde su creación, los computadores han evolucionado:

- Aumentando la velocidad del procesador.
- Aumentando el tamaño de la memoria.
- Disminuyendo el tamaño de los componentes del procesador con lo que disminuye la distancia entre ellos y, por tanto, aumenta la velocidad del procesador.
- Aumentando la capacidad y la velocidad de la E/S (entrada/salida).

#### Organización

También ha habido mejoras que hacen que el procesador siempre esté ocupado:



**Figura 1.3:** La velocidad de los procesadores, según la Ley de Moore, se ha incrementado en un 60 % al año (se duplica cada 1,5 años). En cambio el rendimiento de la memoria DRAM se duplica cada 10 años (se incrementa un 9 % al año).

- Técnicas de ejecución paralela.
- Técnicas de ejecución especulativa.
- Técnicas de encauzamiento.

Cuando se diseña un procesador hay que hacer un balance de las prestaciones de los distintos elementos de forma que las ganancias en prestaciones de un área no perjudiquen a otras áreas. Por ejemplo, la velocidad del procesador ha aumentado más rápidamente que el tiempo de acceso a memoria. Se han utilizado distintas técnicas para compensar este desacople, incluyendo memorias caché, caminos de datos más anchos de la memoria al procesador y más circuitos de memoria inteligentes.

Al observar el gráfico (Fig. 1.3) podemos percatarnos de que existe un desfase entre el rendimiento de la memoria y del procesador, el cual se reduce mediante las diversas organizaciones del procesador.

### Niveles de abstracción

Existen computadores personales, estaciones de trabajo, *mainframes*, supercomputadores, etc. A pesar de la variedad y rapidez de los cambios, se aplican sistemáticamente ciertos conceptos fundamentales. Básicamente, se consideran una serie de niveles desde los que considerar al computador que van desde el nivel superior, donde el usuario ejecuta programas o utiliza el computador, al nivel inferior, consistente en transistores y cables [5]. Además, hay otros niveles intermedios (Fig. 1.4):

- **Físico:** estudio de los componentes fundamentales de un computador (transistores, circuitos integrados, etc). Comprende el estudio de la electrónica, la cual se aborda en las asignaturas de *Física* y *Electrónica* en la titulación de Grado en Ingeniero en Informática de la Universidad de León. El lenguaje máquina se encuentra en este nivel (0s y 1s).
- **Circuito digital:** incluye la electrónica digital, la resolución de circuitos digitales mediante puertas lógicas y circuitos combinacionales y secuenciales, impartida en la asignatura de *Electrónica*.
- **Organización hardware del sistema:** se abordan aspectos como la interconexión entre los distintos componentes que forman el computador (CPU, Memoria, E/S, etc). La organización básica se estudia en *Estructura de Computadores*, mientras que las organizaciones avanzadas son materia de *Arquitectura de Computadores*.
- **Arquitectura del repertorio de instrucciones:** en este nivel se enmarca el lenguaje ensamblador, los registros, etc.
- **Sistema Operativo/Compilador:** se encarga de la gestión de memoria, de los procesos, de ficheros, de la compilación, enlazado, ubicación, etc.
- **Lenguaje de alto nivel:** permite desarrollar aplicaciones. Ejemplos de lenguajes de alto nivel son, entre otros, Pascal, C, Java, C++, etc.
- **Aplicación:** ofimática, comunicaciones, diseño, juegos, etc.

### 1.1.3. Arquitectura de Computadores vs. Estructura de Computadores

El concepto de arquitectura de computadores hace referencia al conjunto de atributos de un computador que son visibles al programador en lenguaje máquina, el sistema operativo o el compilador [5]. Incluye aspectos relacionados con el comportamiento funcional como:

- El tamaño de los diferentes tipos de datos. Por ejemplo, utilizar 16 bits para la representación de enteros.
- Los tipos de operaciones que se pueden realizar: suma, resta, llamadas a subrutinas, etc.

Responde a las preguntas: ¿qué hace?, ¿qué funciones tiene?

La organización hardware del sistema se refiere al conjunto de módulos básicos que componen el computador que, como se verá más adelante, son la CPU, la memoria, la entrada/salida (E/S) y los buses, y a la forma en que se interconectan e interaccionan entre sí [5]. Responde a la pregunta ¿cómo lo hace?

Las unidades funcionales y sus interconexiones dan lugar a especificaciones arquitectónicas. Son relaciones estructurales no visibles al programador como:

- Las interfaces hacia los dispositivos periféricos.



**Figura 1.4:** Niveles desde los que considerar un computador.

- La frecuencia del reloj.
- La tecnología utilizada en las memorias.

Ejemplo:

- Diseño arquitectónico: ¿el computador tendrá la instrucción de multiplicar?, ¿cuánta memoria hay?
- Organización: ¿se implementa con una unidad especializada o con un mecanismo que use la unidad de suma del sistema iterativamente?, ¿qué tecnología de memoria se utiliza?, ¿se emplean cachés distintas para instrucciones y datos?

## 1.2. Perspectiva histórica

### 1.2.1. Era Mecánica

??-s. XVII

Existen máquinas como el ábaco (Fig. 1.5) para la realización de cálculos (sumas y restas). Son de origen oriental y son perfeccionadas por los griegos.



**Figura 1.5:** *El ábaco se utiliza desde la antigüedad.*



**Figura 1.6:** *Calculadora mecánica.*

### 1642-3

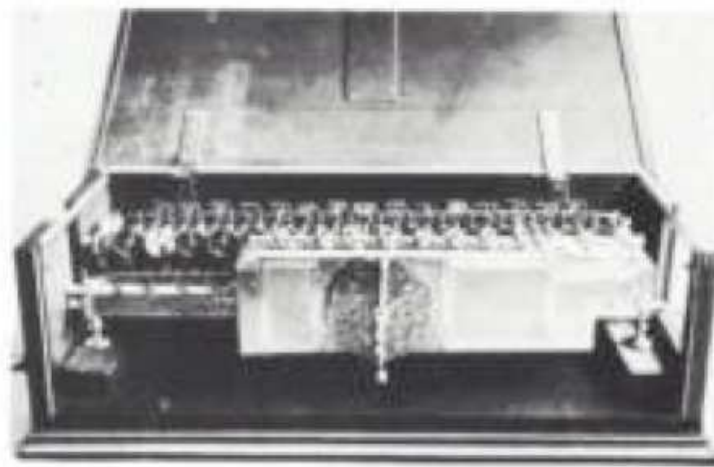
Blas Pascal (Francia) desarrolló una calculadora mecánica (Fig. 1.6) que permitía hacer sumas y restas. Su construcción es difícil y poco fiable.

### 1671

Leibniz desarrolla una máquina que permite multiplicar y dividir (Fig. 1.7).

### 1834-5

Charles Babage (Inglaterra) diseña la *máquina analítica*. No la construye por las restricciones tecnológicas de la época. Sus máquinas tienen funciones básicas presentes en cualquier computador moderno: lectura de datos de entrada, almacenamiento, producción de datos de salida, control automático de secuencia, etc. Tenía mecanismos para realizar bifurcaciones (toma de decisiones), saltos, bucles, subrutinas... Además, se podía programar mediante tarjetas perforadas. Es de propósito general.



**Figura 1.7:** *Máquina de Leibniz.*

#### 1944

Howard Aiken (Harvard) construye un ordenador electromecánico (usando relés) llamado Mark-I (Fig. 1.8). Es de propósito general y es operativo para el cálculo de tablas y la resolución de ecuaciones. Se construyó con la colaboración de IBM (se llama también IBM ASCC). Es programable.

Aunque todas estas máquinas estaban controladas por programa, tenían dos grandes inconvenientes:

- Velocidad de cómputo limitada por la inercia de las partes móviles.
- Transmisión de la información poco fiable y tediosa.

### 1.2.2. Era Electrónica [4, 3]

#### 1904

Fleming inventó la válvula o tubo de vacío diodo [7]. Los primeros diodos eran válvulas o tubos de vacío, también llamadas válvulas termoiónicas constituidas por dos electrodos rodeados de vacío en un tubo de cristal, con un aspecto similar al de las lámparas incandescentes (ver Fig. 1.9a). El invento fue realizado en 1904 por John Ambrose Fleming, de la empresa Marconi, basándose en observaciones realizadas por Thomas Alva Edison. Al igual que las lámparas incandescentes, los tubos de vacío tienen un filamento (el cátodo) a través del cual circula la corriente que produce su calentamiento (efecto Joule). El filamento está tratado con óxido de bario, de modo que al calentarse emite electrones al vacío circundante, electrones que son conducidos electrostáticamente hacia una placa metálica cargada positivamente (el ánodo), produciéndose así la conducción (ver Fig. 1.9b). Evidentemente, si el ánodo no se calienta, no podrá ceder electrones



**Figura 1.8:** Ordenador electromecánico Mark-I.

al vacío circundante, por lo que el paso de la corriente en sentido inverso se ve impedido. Aunque estos diodos aún se emplean en ciertas aplicaciones especializadas, la mayoría de los diodos modernos se basan en el uso de materiales semiconductores, especialmente en electrónica.

### 1906

Lee de Forest crea la válvula de vacío de tres electrodos que resuelve los problemas de los computadores mecánicos [6]. El primer electrodo es el cátodo, que al calentarse produce electrones (Fig. 1.10). El segundo es el ánodo, que está cargado positivamente y, por tanto, atrae a los electrones. El tercero es la rejilla que se sitúa entre el cátodo y el ánodo. La tensión aplicada a la rejilla hace que el flujo de electrones desde el cátodo al ánodo sea mayor o menor. Esto es muy interesante pues aplicando una señal de muy débil intensidad entre el cátodo y la rejilla se puede conseguir que la variación del flujo de electrones entre éste y el ánodo sea muy grande. Es decir, con una pequeña tensión se controla una gran corriente. A este fenómeno se le denomina amplificación. Por ello, el triodo o válvula termoiónica de tres electrodos es un amplificador. Además, se puede utilizar para más funciones, como rectificador o como puertas que dejan pasar la corriente o no (on/off) y que son la base de la electrónica digital.

### 1939

John Atanasoff (Iowa) crea un prototipo de computador digital electrónico que utilizaba aritmética binaria (Fig. 1.11). Era de propósito específico.



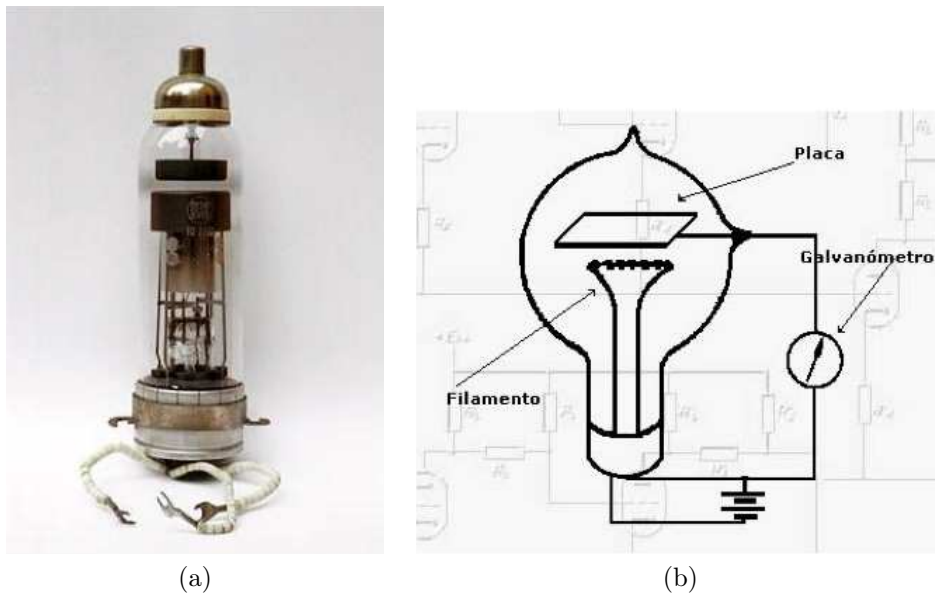


Figura 1.9: Válvula o tubo de vacío diodo.

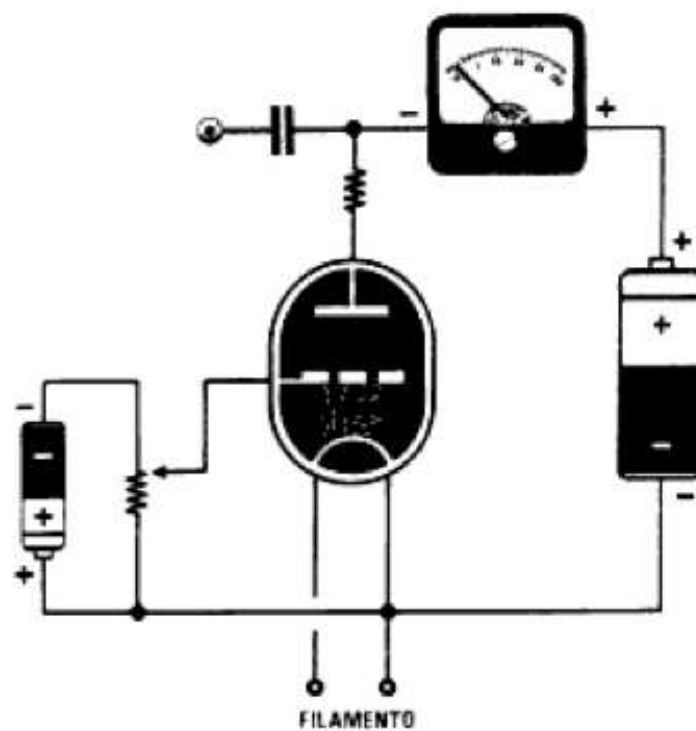


Figura 1.10: Válvula de vacío de tres electrodos.



**Figura 1.11:** *Prototipo de computador digital electrónico creado por John Atanasoff.*

**1ªG**

**1943**

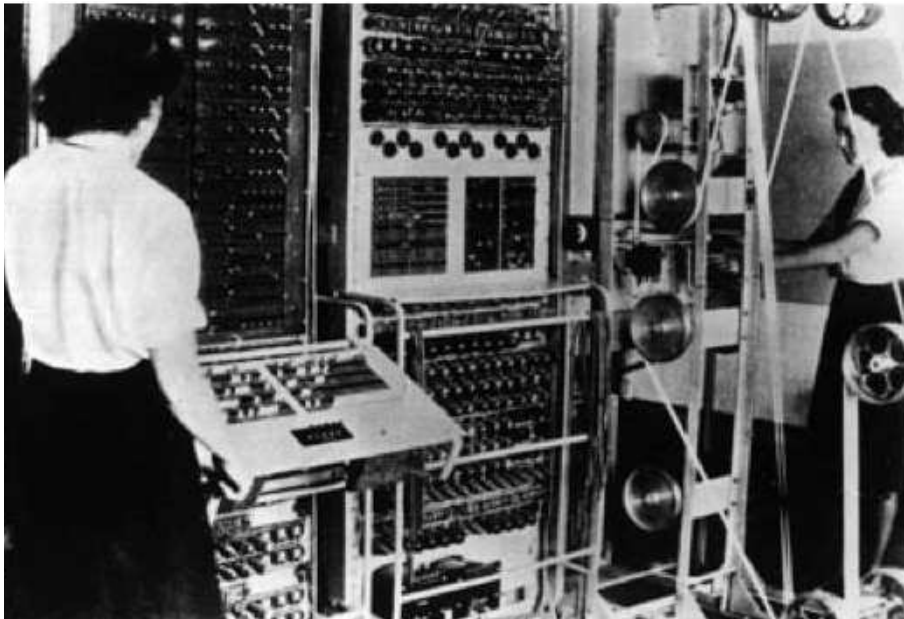
Turing (Inglaterra) y otros construyen la máquina *Colossus* durante la Segunda Guerra Mundial, ver Fig. 1.12. Es completamente electrónica y de propósito específico: descifrar los mensajes secretos de los nazis codificados con la máquina *Enigma*.

**1946 1ª G**

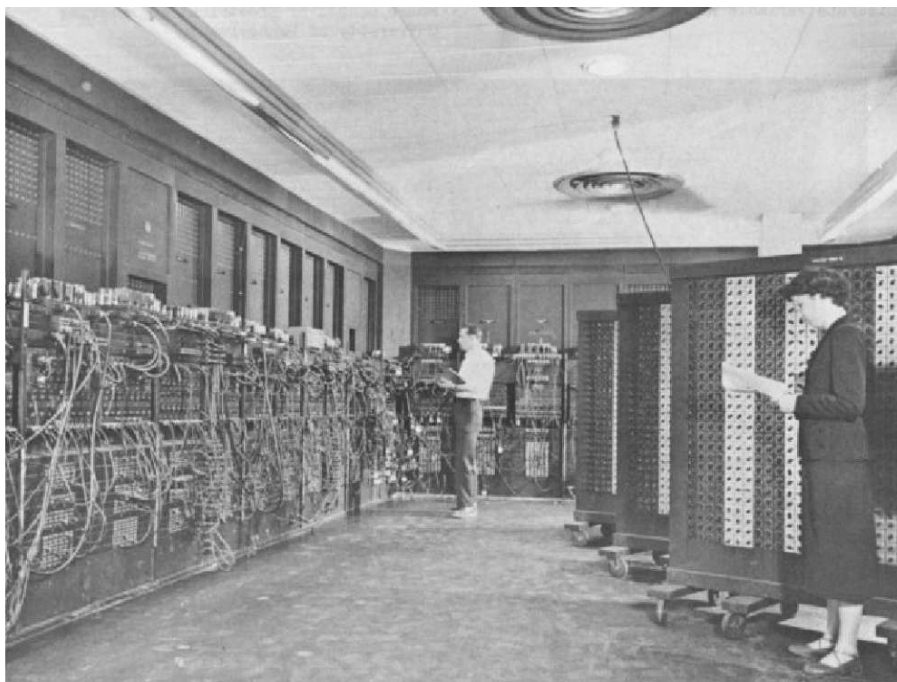
Eckert y Mauchly (Universidad de Pennsylvania) crean el ENIAC (*Electronic Numerical Integrator and Calculator*), el primer computador electrónico de propósito general (Fig. 1.13). Calculaba tablas de trayectorias balísticas para la Marina de EEUU. Tenía 18000 válvulas de vacío y su cálculo era 100 veces más rápido que el de un hombre. Era programable, utilizando cables y conmutadores (el programa no está almacenado) y los datos se introducían mediante tarjetas perforadas. Utiliza aritmética decimal (los números están representados en forma decimal, no en binario).

## CONCEPTO DE PROGRAMA ALMACENADO

Como la tarea de cargar y modificar programas en el ENIAC era tediosa, se piensa que el proceso de programación podría ser más fácil si el programa se representara en una forma adecuada para ser guardado en la memoria junto con los datos. Así, un ordenador podría conseguir



**Figura 1.12:** *Imagen de la máquina Colossus.*



**Figura 1.13:** *ENIAC, primer computador electrónico de propósito general.*



**Figura 1.14:** *EDVAC, computador electrónico de programa almacenado.*

sus instrucciones leyéndolas de la memoria y se podría hacer o modificar un programa colocando los valores en una zona de memoria.

#### **1946-1952 1<sup>º</sup>G**

EDVAC (*Electronic Discrete Variable Computer*), ver Fig. 1.14 es un computador desarrollado por Eckert, Mauchly y John Von Neumann (Moore School). Es un computador de programa almacenado, el programa se guarda codificado como números. Se escribió un informe proponiendo este computador, que sirvió de base para el término de uso habitual *máquina von Neumann*, pero se omitieron los nombres de Eckert y Mauchly.

#### **1946-1949 1<sup>º</sup>G**

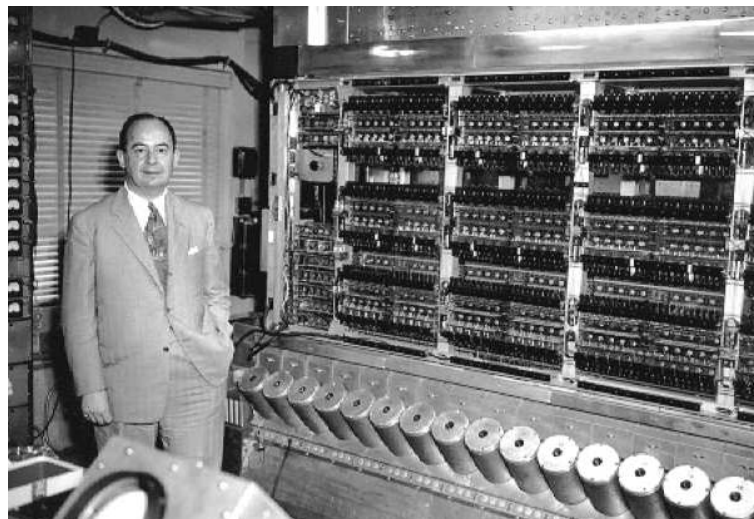
Maurice Wilkes (Universidad de Cambridge) construye el EDSAC (*Electronic Delay Storage Automatic Calculator*) que es el primer computador electrónico de programa almacenado (Fig. 1.15).

#### **1952**

Julian Bigelow construye la máquina IAS (Fig. 1.16) a partir de un informe de Goldstine, von Neumann y A. Burks en el que hablan de la mayoría de los conceptos que se aplican en los ordenadores modernos. Tenía un total de 1024 palabras de 40 bits y era unas 100 veces más rápida que el ENIAC.



**Figura 1.15:** *EDSAC, primer computador electrónico de programa almacenado, contruido por Maurice Wilkes.*



**Figura 1.16:** *Imagen de la máquina IAS contruida por Julian Bigelow.*



**Figura 1.17:** *Mark III, computador construido utilizando válvulas de vacío y con memorias separadas para datos e instrucciones.*

#### 1951-1952

Howard Aiken construyó un par de máquinas de válvulas de vacío, el Mark III (Fig. 1.17) y el Mark IV. A diferencia de otras como el EDSAC, tenían memorias separadas: datos e instrucciones. Hoy en día se emplea el término **ARQUITECTURA HARVARD** para describir máquinas con una sola memoria principal pero con cachés separadas para instrucciones y datos.

#### 1947-

En 1947 se empezó en el MIT el proyecto Whirlwind orientado a aplicaciones de procesamiento de señales de radar en tiempo real; condujo a varias invenciones, siendo la más importante la memoria de núcleo magnético o núcleo de ferrita (la usaban para los registros: 2048 palabras de 16 bits). Los núcleos de ferrita fueron la tecnología de la memoria principal durante casi 30 años. Aparece en el escudo de Ingeniero en Informática (es una estructura toroidal, ver Fig. 1.18).

Cada punto de memoria es un toro o anillo de ferrita que puede presentar dos direcciones de magnetización. Las primeras ferritas fabricadas tenían un diámetro exterior de 3 mm, una capacidad de 2 kbytes y el tiempo de acceso era de unos 30 ms. La conexión de los anillos de ferrita a los transductores se hacía mediante hilos de cobre barnizados que debían hacerse pasar por el interior de las ferritas. Este proceso era de difícil automatización por lo que debía hacerse a mano.



**Figura 1.18:** Escudo de la titulación de Ingeniero en Informática donde aparece la memoria de núcleo magnético o de ferrita.

### 1.2.3. Desarrollos comerciales [4, 3]

Con los computadores electrónicos comienza la producción industrial y comercialización de los mismos.

#### 1949 1ª G

La Eckert-Mauchly Computer Corporation construye su primera máquina, el BINAC (*Binary Automatic Computer*), ver Fig. 1.19.

#### 1951 1ªG

Remington Rand Corp. adquiere Eckert-Mauchly Computer Corporation, que tenía dificultades económicas y es donde se construye el UNIVAC I (*Universal Automatic Computer*), ver Fig. 1.20. Se diseña para ser vendido como ordenador de propósito general. Fue el primer ordenador comercial de éxito: se construyeron 48 unidades y valía un millón de dólares. Incluía un compilador y utiliza aritmética decimal.

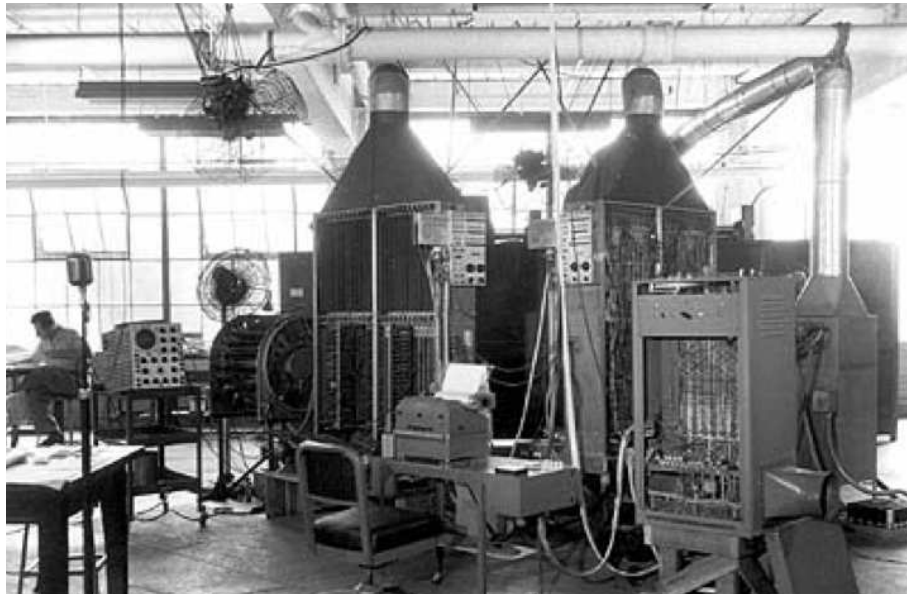
#### 1952 1ª G

IBM había estado en el negocio de las tarjetas perforadas y la automatización de oficinas, pero no empezó a fabricar ordenadores hasta 1950. El IBM 701 (Fig. 1.21) es el primer ordenador de IBM. Salió al mercado en 1952 y se vendieron 19 unidades.

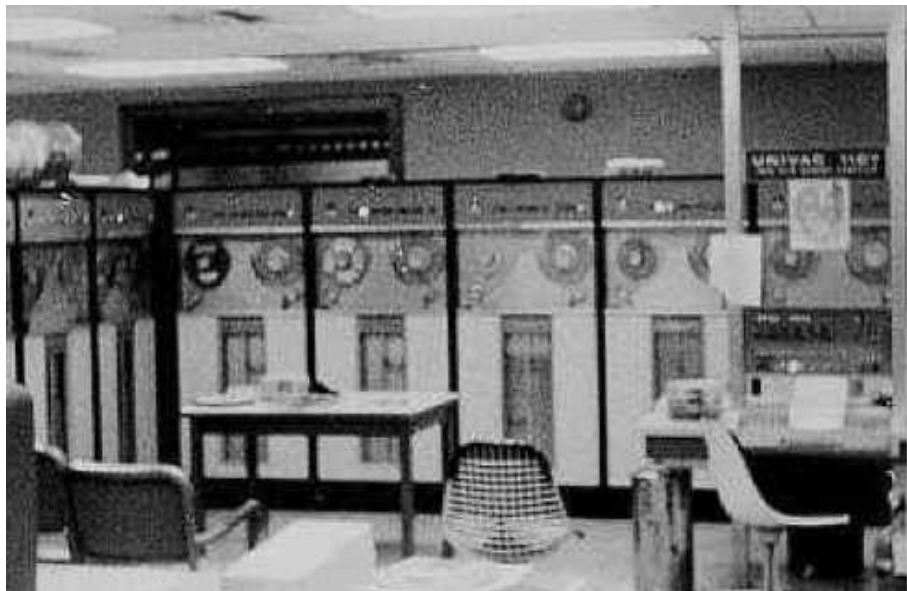
**2ªG** En los 50 se pensaba que el mercado y las posibilidades para estas máquinas “altamente especializadas” eran muy limitadas y auguraban un futuro pesimista.

#### 1964 3ªG

IBM anuncia 6 implementaciones de la arquitectura Sistema/360 (Fig. 1.22) que variaban en precio y rendimiento. Aparece el concepto de **FAMILIA DE COMPUTADORES** que es un



**Figura 1.19:** *BINAC, la primera máquina construída por la Eckert-Mauchly Computer Corp.*



**Figura 1.20:** *UNIVAC, el primer ordenador comercial de éxito.*





**Figura 1.21:** *IBM 701, el primer ordenador de IBM.*

conjunto de computadores con:

- Similar arquitectura.
- Distinta organización.

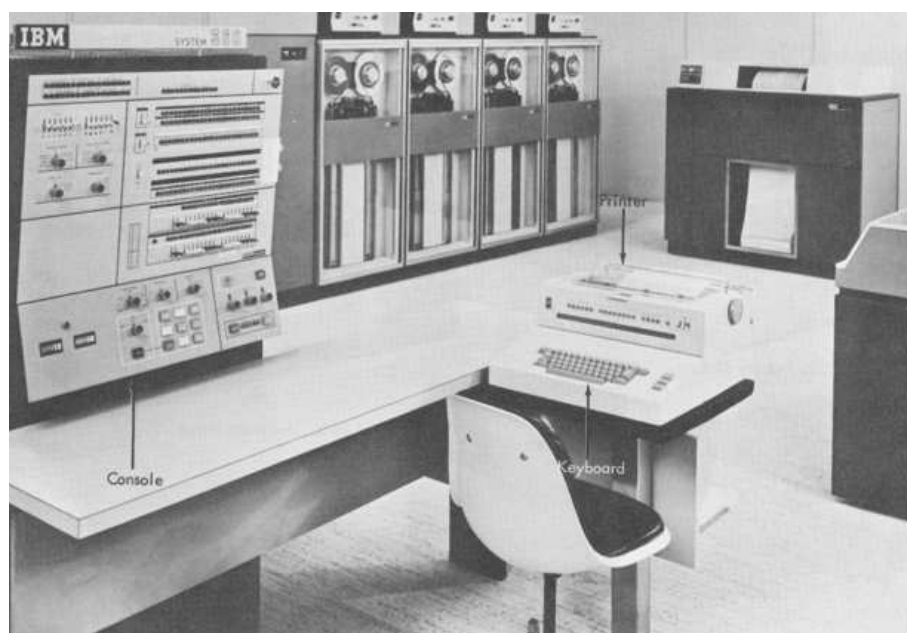
Las familias de computadores hacen posible que:

- Existan máquinas de la misma familia con distinta tecnología, velocidad, prestaciones y precio.
- Las máquinas de una misma familia sean compatibles entre sí. Todos los miembros de una misma familia pueden ejecutar los mismos programas.

La compatibilidad suele ser sólo hacia arriba (*upward compatibility*), es decir, las máquinas de gama alta de una familia pueden ejecutar los mismos programas que las máquinas de gama baja, pero lo contrario no suele ser cierto.

Ejemplos de familias:

- Familia PC compatible: IBM PC-XT (8086), IBM PC-AT (80286), 80386, 80486, Pentium, Pentium Pro, Pentium II, Pentium III, Pentium IV.
- Familia Motorola 68XXX: MC68000, MC68010, MC68020, MC68030, MC68040, MC68060.
- Familia PowerPC: 601, 603, 604, 620, 740/750, G3, G4, G5.



**Figura 1.22:** Sistema/360, origen del concepto familia de computadores.

### 1965 3<sup>a</sup>G

Digital Equipment Corporation (DEC) da a conocer el PDP-8, el primer miniordenador comercial (Fig. 1.23). Supone un gran avance en el diseño a bajo coste (por debajo de los 20000\$). Los miniordenadores son precursores de los microprocesadores.

### 1971 3<sup>a</sup>G

Intel introduce el microprocesador Intel 4004 (Fig. 1.24). Es el primer chip que contiene todos los componentes de la CPU en un solo chip. Luego comercializó el 8008, 8080, ..., 80x86, ...

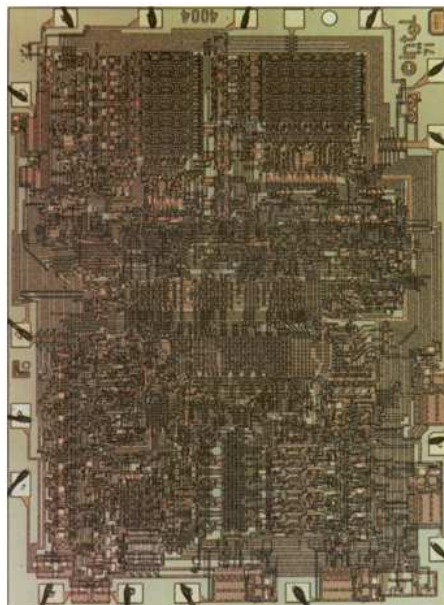
En cuanto a los supercomputadores, señalar que el primero fue el CDC6000 diseñado por Seymour Cray, de la Control Data Corporation en 1963, ver Fig. 1.25 (3<sup>a</sup>G). Luego fundó la Cray Research Inc. y en 1976 anunció el Cray-1 (Fig. 1.26): fue la máquina más rápida del mundo, la más cara y el computador con la mejor relación coste/rendimiento para programas científicos. En 1996 la Cray Research pasó a manos de Silicon Graphics (hoy Cray Inc.).

Otros diseñadores estudiaban la posibilidad de utilizar el microprocesador de forma que se pudiera tener un ordenador en casa. No hay un único origen para los ordenadores personales, pero en 1977 el Apple II de Steve Jobs y Steve Wozniak (Fig. 1.27a) sentó las líneas maestras del bajo coste, alto volumen y alta fiabilidad, que definieron la industria del ordenador personal (4<sup>a</sup>G).

El IBM Personal Computer anunciado en 1981 llegó a ser el más vendido de todos los ordenadores de cualquier tipo (Fig. 1.27b).



**Figura 1.23:** *Primer miniordenador comercial, el PDP-8.*



**Figura 1.24:** *Intel se dedicó a comercializar los microprocesadores, chips que contienen todos los componentes de la CPU.*



**Figura 1.25:** *El supercomputador CDC6000.*



**Figura 1.26:** *El supercomputador Cray-1.*



(a)



(b)

**Figura 1.27:** *Ejemplos de ordenadores personales: el Apple II (a) y el IBM Personal Computer (b).*

	Tecnología	Nuevo producto principal
<b>1ª Generación</b> (1950-59)	Válvulas de vacío	Ordenador electrónico comercial UNIVAC I, IBM 701
<b>2ª Generación</b> (1960-68)	Transistores	Ordenador más baratos IBM 7070, CDC 6600
<b>3ª Generación</b> (1969-77)	Circuito integrado	Minicomputadores IBM 360, Cray-1, PDP-8 y PDP-11/20
<b>4ª Generación</b> (1978-)	LSI Y VLSI	Ordenadores personales y estaciones de trabajo Pentium, R10000, PowerPC

**Tabla 1.1:** *Generaciones de ordenadores.*

El procesador MIPS, el cual vamos a tratar en esta asignatura, se utiliza en computadores, pero principalmente está integrado en consolas de videojuegos (la Nintendo 64 tiene un MIPS R4399 de 64 bits), impresoras láser, etc. La PDA Cassiopeia E-125 (o la EM-500) tienen un MIPS VR4122 a 150 MHz.

#### 1.2.4. Generaciones de ordenadores

Desde 1952 han ido apareciendo miles de computadores nuevos con una amplia variedad de tecnologías y capacidades. Se clasifican en generaciones, en función de la tecnología utilizada. Por convención, se considera 1ª generación a la primera de ordenadores electrónicos comerciales, sin tener en cuenta las máquinas mecánicas o electromecánicas que les precedieron.

En la 1ª generación de computadores las instrucciones se procesaban en serie: se buscaba la instrucción, se decodificaba y luego se ejecutaba. La velocidad típica de procesamiento que alcanzaron los computadores era aproximadamente 40000 operaciones por segundo. Eran equipos de gran tamaño, escasa capacidad y difícil mantenimiento que disipaban mucho calor. Los trabajos se realizaban en monoprogramación y no existía sistema operativo, por lo que los periféricos de Entrada/Salida dependían directamente del procesador. Se programaba en lenguaje máquina lo que exigía programadores muy especializados.

#### 1.2.5. Ley de Moore

El cofundador de Intel, Gordon Moore, pronosticó en 1965 que la cantidad de transistores incorporados en los microprocesadores sería duplicada cada 8 meses (Fig. 1.28). La previsión de Moore resultó ser tan exacta que su teoría pasó a denominarse *Ley de Moore*.

### 1.3. Modelo von Neumann

Este modelo sirve como base para la mayoría de los computadores comerciales actuales.

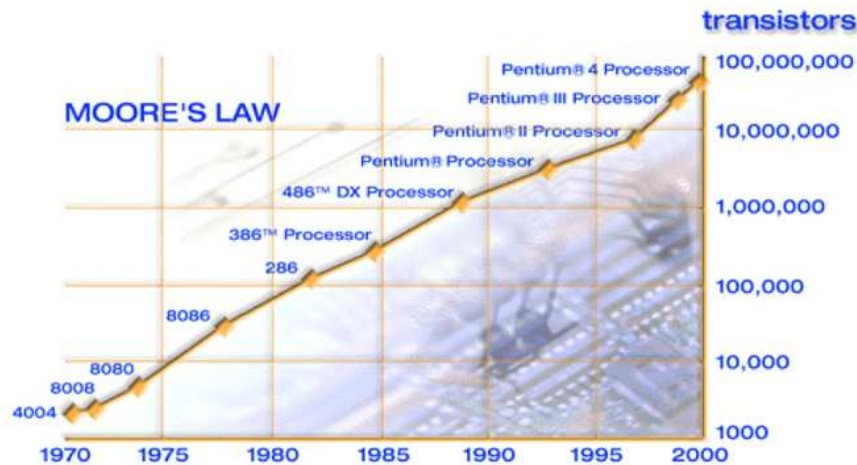


Figura 1.28: Ley de Moore.

### 1.3.1. Estructura

Consta de 5 componentes principales (ver Fig. 1.29) [2]:

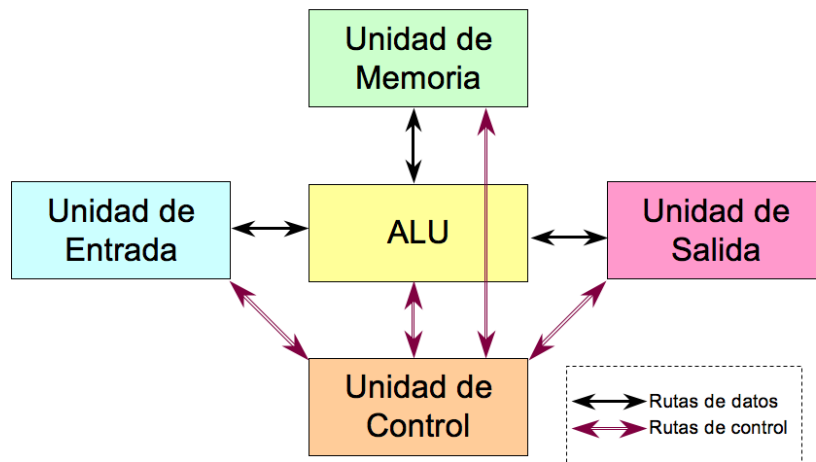
- **Unidad de Entrada:** provee las instrucciones y los datos requeridos por el sistema.
- **Unidad de Memoria:** donde se almacenan.
- **Unidad Aritmético-Lógica (ALU):** se procesan las instrucciones y los datos.
- **Unidad de Control:** dirige ese proceso.
- **Unidad de Salida:** a donde se envían los resultados obtenidos.

El conjunto constituido por las unidades aritmético-lógica y de control se designa habitualmente bajo el nombre de **Unidad Central de Proceso (CPU)**.

### 1.3.2. Características principales

La arquitectura de von Neumann se fundamenta en tres ideas claves [2]:

- Su funcionamiento se basa en el concepto de **programa almacenado en memoria**. Antes no se almacenaba el programa junto con los datos a procesar sino en un medio externo: tableros de clavijas, cintas o tarjetas perforadas. En el computador de programa almacenado, el programa puede manipularse como si se tratara de datos. Este concepto da origen a los compiladores y sistemas operativos y es la base de la gran versatilidad de los computadores modernos. Por tanto, en la memoria del computador se almacenan:
  - *Instrucciones:* programa que controla el funcionamiento de un computador.
  - *Datos:* que procesa y genera dicho programa.



**Figura 1.29:** Esquema del modelo von Neumann en el que se indican las rutas que siguen los datos y las líneas de control.

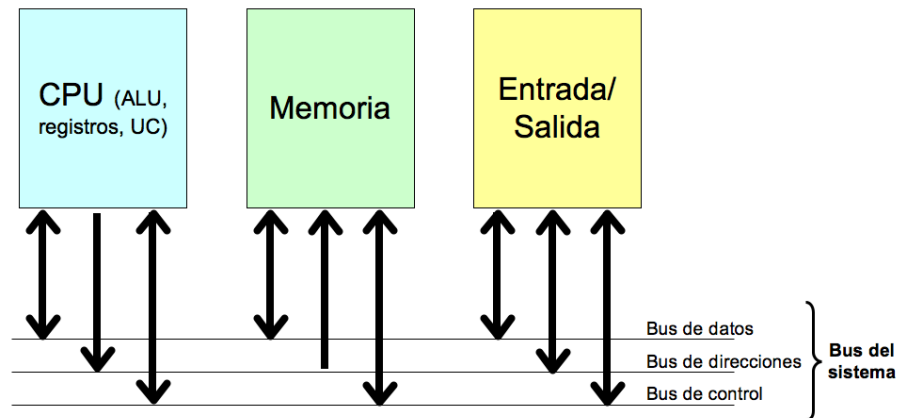
	...		
0xFF3B	21360	⇔	a=21360;
	...		
	...		
0xFFF4	21360	⇔	b=b+1;
	...		

**Tabla 1.2:** Ejemplo de memoria.

- Se puede acceder a la información contenida en la memoria especificando la dirección donde se encuentra almacenada. La memoria no distingue si lo que contiene es un dato o una instrucción (ver Tabla 1.2). El que sea considerado de una u otra forma depende únicamente de quién recibe la información.
- La **ejecución** de un programa (conjunto de instrucciones) es por defecto **secuencial**. Se pasa de una instrucción a la que le sigue inmediatamente a no ser que explícitamente se modifique esta secuencia mediante una ruptura (salto) ya sea condicional o incondicional.

### 1.3.3. Modelo de interconexión a través del bus

Aunque el modelo von Neumann prevalece en la estructura de los computadores actuales, se ha modernizado, considerando un modelo que utiliza el sistema de interconexión a través



**Figura 1.30:** Esquema del modelo de interconexión a través del bus del sistema. Como la CPU no tiene unidades direccionables, nunca le llegan direcciones. Por otro lado, la memoria nunca “pide” información de una unidad direccionable (es un componente pasivo).

de lo que se denomina **bus del sistema** [2]. Se considera que el sistema de computación está constituido por tres subconjuntos (Fig. 1.30):

- La **CPU**: combina la ALU y la Unidad de control en un solo bloque funcional.
- La **Memoria**.
- La **Entrada/Salida**: combinando la unidad de entrada y la de salida.

La comunicación entre componentes se realiza por medio de un camino compartido conocido como bus del sistema, constituido a su vez por:

- Un **bus de datos**: transporta la información que se está transmitiendo.
- Un **bus de direcciones**: determina hacia dónde está siendo enviada dicha información.
- Un **bus de control**: describe aspectos de la forma en que se está llevando a cabo la mencionada transferencia de información.

Existe también un bus de alimentación que lleva energía eléctrica a los componentes. Algunas arquitecturas pueden tener, además de los anteriores, un bus de entrada/salida.

Los buses están constituidos por conjuntos de cables agrupados de acuerdo con su función. Un bus de datos de 32 bits contiene 32 cables individuales cada uno de los cuales transporta un bit de datos (distinguiéndolo de la información de direcciones o control). En este sentido, el bus del sistema es, en realidad, un grupo de buses individuales clasificados de acuerdo con su función.



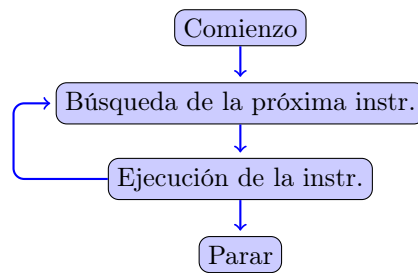


Figura 1.31: Ciclo básico de una instrucción.

## 1.4. Función del computador

Hasta ahora se ha visto cómo es un computador y en esta sección se aborda qué hace. La función básica que realiza un computador es la **ejecución de un programa** [1]. Como se ha visto, un programa consiste en un conjunto de instrucciones y datos almacenados en la unidad de memoria. La CPU se encarga de ejecutar las instrucciones especificadas en el programa.

### 1.4.1. Ciclo de instrucción

La secuencia de operaciones realizadas en la ejecución de una instrucción constituye lo que se denomina **ciclo de instrucción** [1].

### 1.4.2. Fases de ejecución de una instrucción

El ciclo de instrucción consta de dos pasos principales (ver Figura 1.31):

1. Fase o ciclo de búsqueda de la instrucción. Lo que se hace es:

- Se localiza la instrucción en la memoria.
- Se guarda la instrucción en el Registro de Instrucción (IR).
- Se lee la instrucción.
- Se actualiza el valor del PC.

2. Fase o ciclo de ejecución de la instrucción. Generalmente:

- Se decodifica la instrucción.
- Se buscan los operandos.
- Se opera según el código de la instrucción.

La ejecución de un programa consiste en la repetición cíclica de las fases de búsqueda y ejecución de las instrucciones de que consta y finaliza mediante una de las siguientes acciones:

- Si se desconecta el computador.

	...
0x230	Instr1
0x234	Instr2
0x238	Instr3
0x23C	Instr4
	...

**Tabla 1.3:** *Ejemplo de instrucciones de 4 bytes almacenadas en memoria.*

- Si ocurre algún tipo de error irrecuperable.
- Si se ejecuta en el propio programa una instrucción que fuerza su terminación.

En el comienzo de cada ciclo de instrucción, la CPU busca en la memoria una instrucción. En una CPU tipo von Neumann se dispone de un registro especial llamado **contador de programa** (PC) que se utiliza para llevar el control de la posición de memoria donde tiene que ir a buscar la próxima instrucción. A menos que se le indique lo contrario, la CPU siempre incrementa el PC después de cada fase de búsqueda de la instrucción (la ejecución es secuencial por defecto).

#### Ejemplo.

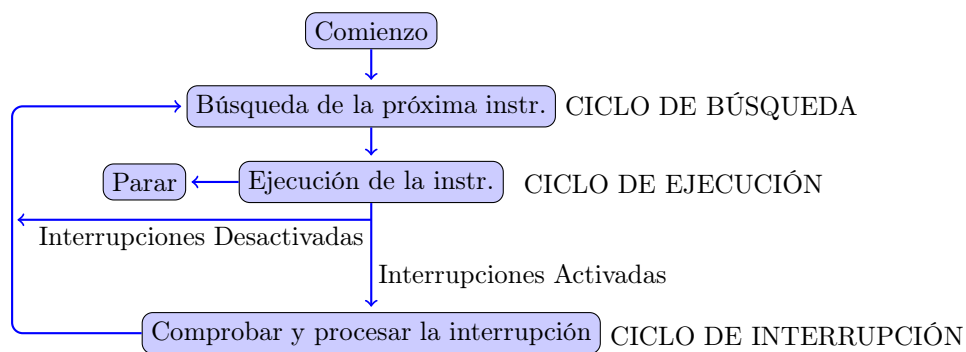
Supongamos instrucciones de 32 bits (4 bytes) que están almacenadas en memoria según aparece en la Tabla 1.3. ¿Qué valor tiene el PC cuando se está ejecutando la instrucción 2?

**Solución:** PC=0x238

### 1.4.3. Ciclo de interrupción

La mayoría de los dispositivos periféricos son mucho más lentos que la CPU, así que si ésta transfiere datos a, por ejemplo, una impresora con el ciclo de instrucción anterior, después de cada operación de escritura la CPU tendrá que parar y permanecer en un estado de espera hasta que la impresora finalice su tarea. Esto es un mal uso de la CPU. Para que la CPU ejecute otras instrucciones mientras se está realizando la operación de E/S se utilizan las interrupciones [1].

Supongamos que la CPU tiene dos programas, uno que lee datos de la memoria y los transfiere a un dispositivo periférico y el otro una aplicación numérica que necesita una gran cantidad de cálculo. La CPU inicia el primer programa emitiendo una orden de escritura al dispositivo periférico y, a continuación, comienza la ejecución de la otra aplicación. Si el dispositivo de salida está ya preparado para continuar (puede aceptar más datos de la CPU), el módulo de E/S para ese periférico envía a la CPU una señal de petición de interrupción. La CPU responde suspendiendo la operación del programa que tiene en ejecución y activando el programa que presta servicio a ese dispositivo particular de E/S. Al terminar de atender esa petición, la CPU vuelve otra vez al programa original y continúa su ejecución en el punto en el que lo interrumpió. El ciclo de instrucción con interrupciones sería el que aparece en la figura 1.32.



**Figura 1.32:** *Ciclo de instrucción con interrupciones.*

En el ciclo de interrupción, la CPU comprueba si se ha producido alguna interrupción. Si no hay ninguna, continúa con el ciclo de búsqueda de la siguiente instrucción del programa que tiene en ejecución. En cambio, si está activada la señal de interrupción (hay al menos una interrupción pendiente de ser atendida):

- Se guarda el contexto del programa que en ese momento se está ejecutando para poder continuar después la ejecución en el punto donde lo interrumpió. Se guarda el contenido del PC y el resto de registros. En el PC está la dirección de la próxima instrucción que se va a ejecutar.
- Carga en el contador de programa la dirección de comienzo de un programa encargado de gestionar la interrupción.

La CPU realiza el ciclo de búsqueda: lee la primera instrucción del programa que gestiona la interrupción y se encarga de dar servicio a la petición que estaba solicitada y pendiente.

Todos los sistemas operativos incorporan un programa gestor de interrupciones que analiza la naturaleza de la interrupción y realiza las acciones necesarias. Al acabar, la CPU continúa con la ejecución en el punto donde la dejó. Aunque hay un cierto retardo, este mecanismo permite un uso mucho más eficaz de la CPU.

Las interrupciones no se atienden siempre de forma instantánea. Se pueden desactivar todas o algunas y posteriormente volverlas a activar. Una interrupción desactivada significa que la CPU ignorará esa señal de petición de interrupción. Por ejemplo, generalmente interesa finalizar el procesamiento de una interrupción antes de atender a otra, por lo que el mecanismo de interrupciones a menudo se desactiva cuando la CPU está atendiendo a una de ellas. Si durante ese tiempo ocurre una nueva petición de interrupción, quedará pendiente y será comprobada por la CPU una vez que haya vuelto a activar el mecanismo de interrupciones.

Las acciones que realiza la CPU cuando atiende una interrupción son:

- Suspende la ejecución del programa en curso y salva el contenido del PC y de los otros registros.



**Figura 1.33:** *Formato de instrucción.*

- Inhibe las interrupciones y ejecuta el programa de servicio de interrupción.
- Se informa al periférico de que ha atendido su interrupción (petición).
- Finalizado el programa de servicio, se activa el sistema de interrupciones.
- Se recupera el PC y los registros y continúa con el programa interrumpido en el punto en que se dejó.

En un sistema con interrupciones encadenadas (*daisy chain*), la prioridad de un determinado módulo depende de su posición en la cadena.

#### Problema [1].

Supongamos que tenemos una máquina cuyas instrucciones están formadas por 32 bits y tienen dos campos (Figura 1.33):

- Código de operación: formado por los 6 primeros bits.
- Dirección: formado por el resto de bits.

La CPU tiene los siguientes registros:

- Contador de programa (PC): guarda la dirección de la instrucción.
- Registro de Instrucción (IR): guarda la instrucción que se está ejecutando.
- Acumulador (AC): registro de almacenamiento temporal.

Se pueden realizar las siguientes operaciones, que se indican con los distintos valores que toma el código de la operación:

- 000001: cargar el AC desde memoria.
- 000011: cargar el AC desde E/S.
- 000010: almacenar el AC en memoria.
- 000111: almacenar el AC en E/S.
- 001111: sumar el AC de memoria.

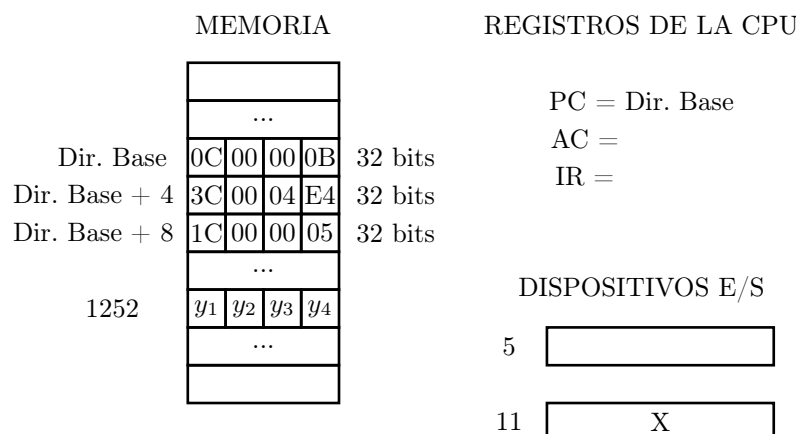
En las operaciones de E/S, los 26 bits del campo dirección identifican un dispositivo concreto de E/S. Se pide mostrar la evolución de las partes relevantes de la memoria, de los dispositivos de E/S y de los registros de la CPU, en la ejecución del siguiente programa:

1. Cargar AC desde el dispositivo 11.
2. Sumar el contenido de la posición de memoria 1252.
3. Almacenar el contenido del AC en el dispositivo 5.

**Solución:** Primer paso: codificar las instrucciones que componen el programa, es decir, determinar para cada una de ellas cuál es el valor de sus campos. Se utiliza notación hexadecimal. Para ello, se agrupan los bits de 4 en 4, de derecha a izquierda y se sustituye cada grupo de 4 bits por el dígito hexadecimal equivalente:

1. Instrucción1: Cargar AC desde el dispositivo 11.
  - Código de operación: Según la tabla, el valor es 000011.
  - Dirección: El número 11 en binario con 26 bits corresponde con 0...0001011.
  - Instrucción entera: 0000 1100 0000 0000 0000 0000 0000 1011<sub>2</sub> = 0X0C00000B.
2. Instrucción2: Sumar el contenido de la posición de memoria 1252.
  - Código de operación: Según la tabla, el valor es 001111.
  - Dirección: El número 1252 en binario con 26 bits corresponde con 0...00010011100100.
  - Instrucción entera: 0011 1100 0000 0000 0000 0100 1110 0100<sub>2</sub> = 0X3C0004E4.
3. Instrucción3: Almacenar el contenido del AC en el dispositivo 5.
  - Código de operación: Según la tabla, el valor es 000111.
  - Dirección: El número 5 en binario con 26 bits corresponde con 0...0000101.
  - Instrucción entera: 0001 1100 0000 0000 0000 0000 0000 0101<sub>2</sub> = 0X1C000005.

En el enunciado del problema no se menciona a partir de qué posición concreta de memoria se encuentran almacenadas estas tres instrucciones. Por tanto, se puede suponer que se trata de una posición genérica a la que se llamará *dirección\_base*. Si se quiere que la máquina propuesta en el enunciado empiece a ejecutar la primera instrucción, PC tiene que tener el valor de esa dirección. Tampoco se conoce el valor que suministra el dispositivo 11 de E/S referenciado en la primera instrucción, ni el contenido inicial de la posición de memoria 1252 utilizado en la segunda instrucción. Supongamos que son los valores *X* e *Y*, respectivamente. Por tanto, la situación inicial sería la que aparece en la figura 1.34.



**Figura 1.34:** Situación inicial.

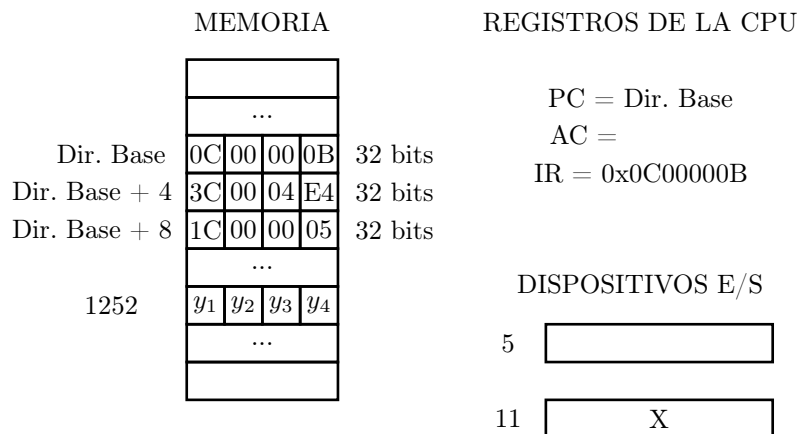
En la fase de búsqueda la instrucción contenida en la dirección de memoria que indique el PC se lleva al IR. Esta fase es igual para todas las instrucciones. En la fase de ejecución, aunque depende de la naturaleza de la instrucción que se va a ejecutar, normalmente requiere la decodificación de la instrucción, la búsqueda de los operandos y la ejecución de la operación especificada por el código de operación.

1. Cargar AC desde el dispositivo 11.

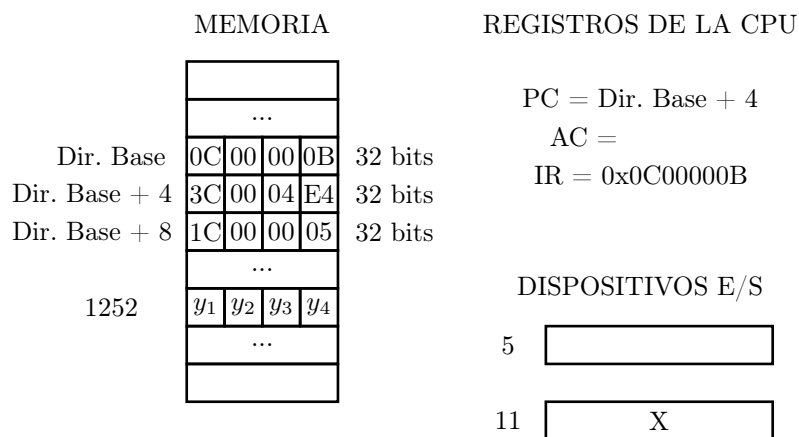
- PC = *dirección\_base*; que es la dirección de la primera instrucción del programa y, por tanto, el valor almacenado en esta dirección de memoria se carga en el registro de instrucción (IR), como se muestra en la Figura 1.35.
- Una vez que se ha leído la instrucción, se actualiza el PC para que apunte a la instrucción siguiente (Fig. 1.36). Esta sería la situación al final de la fase de búsqueda de la primera instrucción.
- Se decodifica la instrucción. Los 6 primeros bits indican que hay que cargar el acumulador (AC) desde un dispositivo de E/S, y los 26 bits restantes indican que dicho dispositivo es el número 11 (Figura 1.37). En dicha Figura se muestra cómo sería la situación después de la fase de ejecución de la primera instrucción.

2. Sumar el contenido de la posición de memoria 1252.

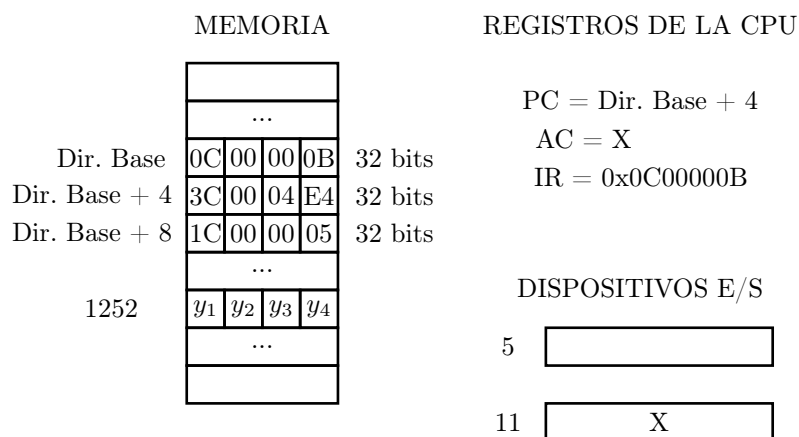
- Se lee la instrucción donde apunta el PC (*Dir. Base + 4*) y se lleva de memoria al IR (Figura 1.38).
- Se actualiza el PC para que apunte a la instrucción siguiente. En la Figura 1.39 vemos cómo queda la máquina después de la fase de búsqueda de la segunda instrucción.



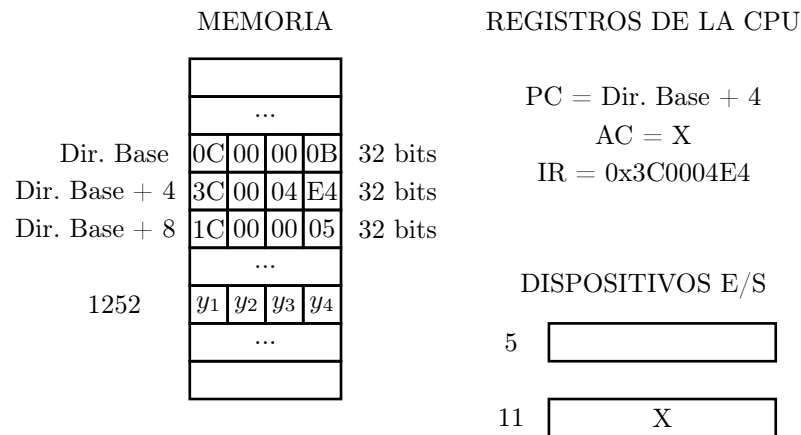
**Figura 1.35:** El valor almacenado en la posición de memoria donde apunta el PC se carga en el registro de instrucción IR.



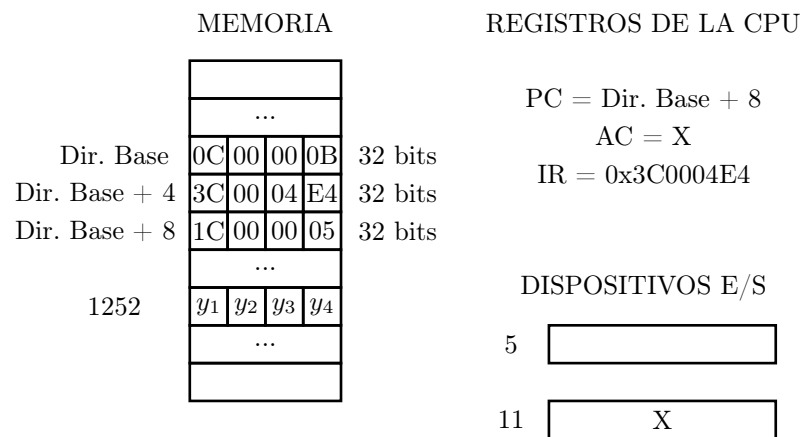
**Figura 1.36:** El PC apunta a la siguiente instrucción.



**Figura 1.37:** Se carga el acumulador AC desde el dispositivo de E/S número 11.



**Figura 1.38:** Se lee la segunda instrucción.



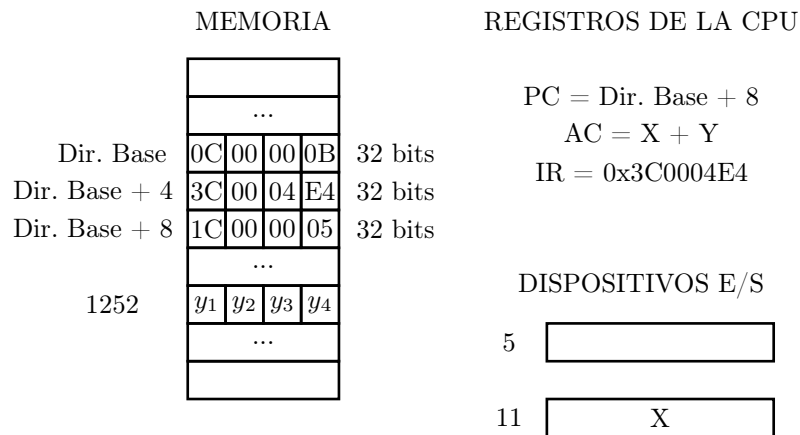
**Figura 1.39:** Se incrementa el PC para que apunte a la tercera instrucción.

- c) Se decodifica la instrucción. Al contenido del AC se le suma el contenido de la posición de memoria 1252, utilizando algún recurso de cálculo que necesariamente debe poseer la máquina. El resultado se almacena en el AC. La Figura 1.40 representa el estado de la máquina después de la fase de ejecución de la segunda instrucción.

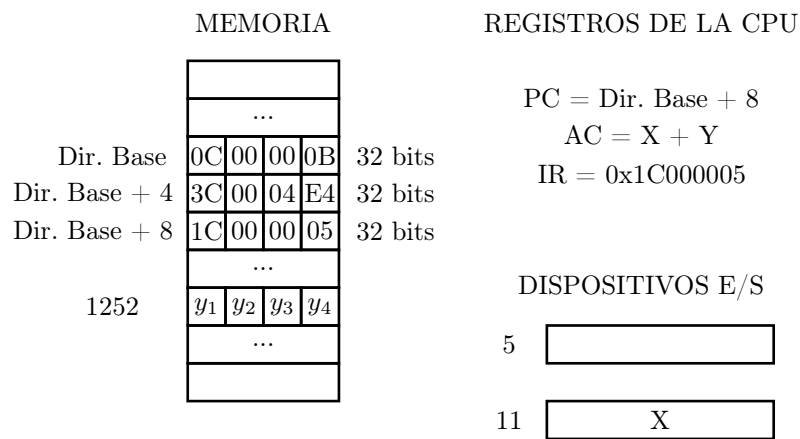
### 3. Almacenar el contenido del AC en el dispositivo 8.

- a) Se lleva la instrucción donde apunta el PC de memoria al IR (Figura 1.41).
- b) Se actualiza el PC para que apunte a la instrucción siguiente. En la Figura 1.42 se muestra el estado de la máquina después de la fase de búsqueda de la tercera instrucción.
- c) Se decodifica la instrucción y se almacena el contenido del AC en el dispositivo número 5. Después de la fase de ejecución de la tercera instrucción, la máquina queda como



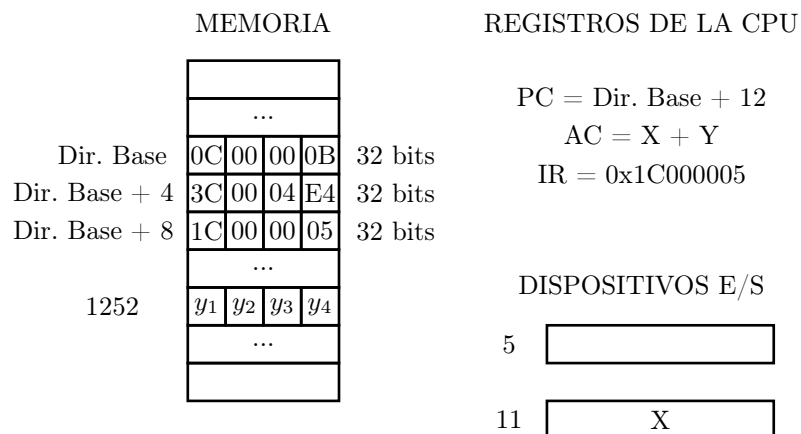


**Figura 1.40:** Se almacena en el AC el resultado de sumar el contenido de la posición de memoria 1252 con el contenido del AC.

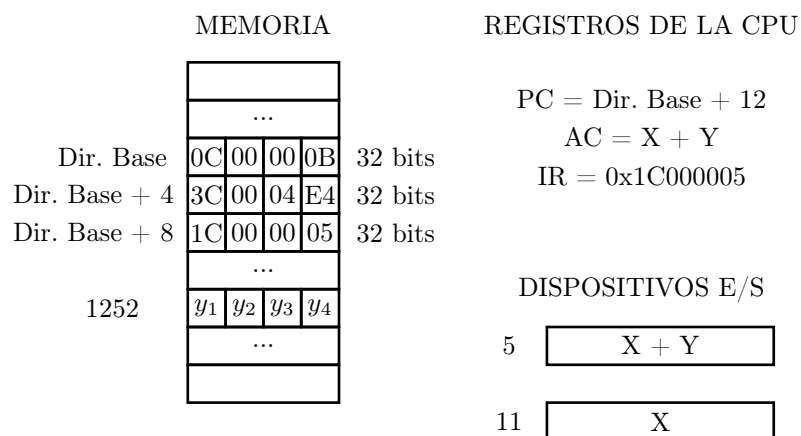


**Figura 1.41:** Se lee la última instrucción y se carga en el IR.

se muestra en la Figura 1.43.



**Figura 1.42:** Se actualiza el valor del PC.



**Figura 1.43:** Se almacena el AC en el dispositivo 5.

# Bibliografía

- [1] Dormido, S., Canto M.A. Mira J. Delgado A.E. 2002. *Estructura y Tecnología de Computadores*. Sanz y Torres. 1.4, 1.4.1, 1.4.3, 1.4.3, 2.4, 2.4.1
- [2] Murdocca, M.J., Heuring V.P. 2002. *Principios de Arquitectura de Computadores*. Prentice Hall. 1.3.1, 1.3.2, 1.3.3, 2.2, 2.3.5
- [3] Patterson, D.A., Hennessy J.L. 1997. *Computer Organization and Design*. Morgan Kaufmann. (document), 1.1.1, 1.1.2, 1.2.2, 1.2.3, 2.3.1, 2.3.2, 2.3.3, 2.3.3, 2.3.3, 2.3.4
- [4] Patterson, D.A., Hennessy J.L. 2000. *Estructura y Tecnología de Computadores*. Reverté. (document), 1.1.1, 1.1.2, 1.2.2, 1.2.3, 2.3.1, 2.3.2, 2.3.3, 2.3.3, 2.3.3, 2.3.4
- [5] Stallings, W. 2000. *Organización y Arquitectura de Computadores*. Prentice Hall. 1.1.1, 1.1.2, 1.1.2, 1.1.3
- [6] Wikipedia. 2009a. *Triodo* — *Wikipedia, La enciclopedia libre*. [Internet; descargado 15-febrero-2009]. 1.2.2
- [7] Wikipedia. 2009b. *Válvula termoiónica* — *Wikipedia, La enciclopedia libre*. [Internet; descargado 15-febrero-2009]. 1.2.2