Arquitectura i Configuracions Informàtiques

Tema 1. Introducción – Parte 2

Davide Careglio

Introducción

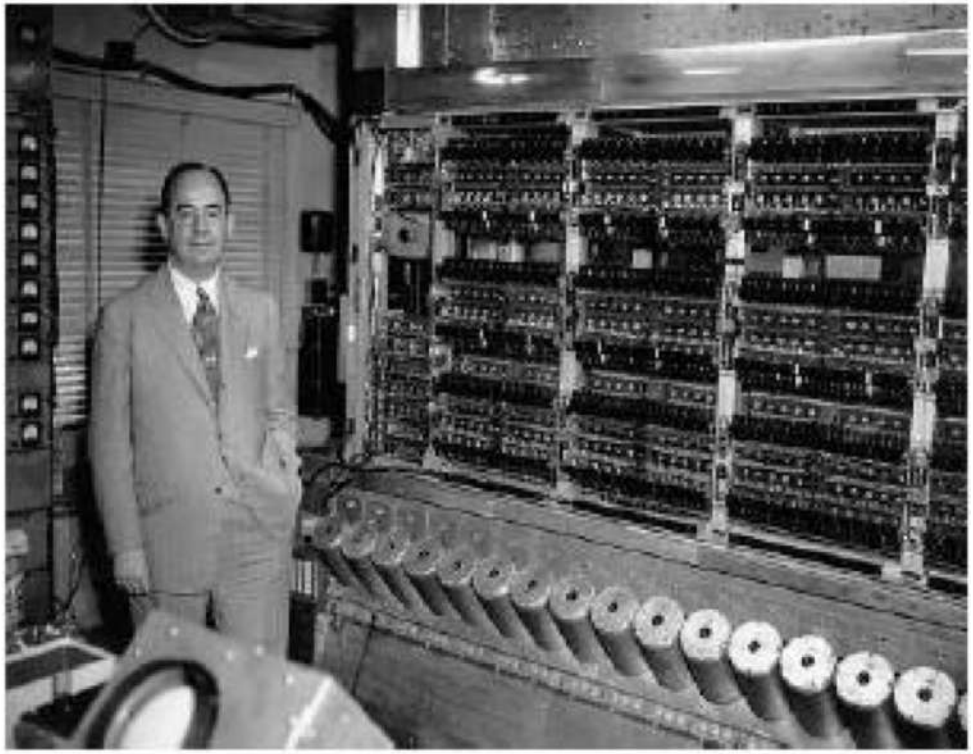
- Tema 1. Introducción
- Tema 2. El microprocesador
- Tema 3. Memoria
- Tema 4. Dispositivos de E/S y buses
- Tema 5. DataCenters y modelos de comunicación

Tema 1. Introducción

- Tema 1. Introducción
  - Un poco de historia
  - Generaciones
  - Arquitectura de los ordenadores
  - Identificación de los componentes
- Tema 2. El microprocesador
- Tema 3. Memoria
- Tema 4. Dispositivos de E/S y buses
- Tema 5. DataCenters y modelos de comunicación

1.3 - Arquitectura

- Arquitectura de Von Neumann
- En 1945, el matemático Janos von Neumann presenta los principios generales que debe seguir una máquina de propósito general.
- Define
  - Estructura básica
  - Comportamiento funcional



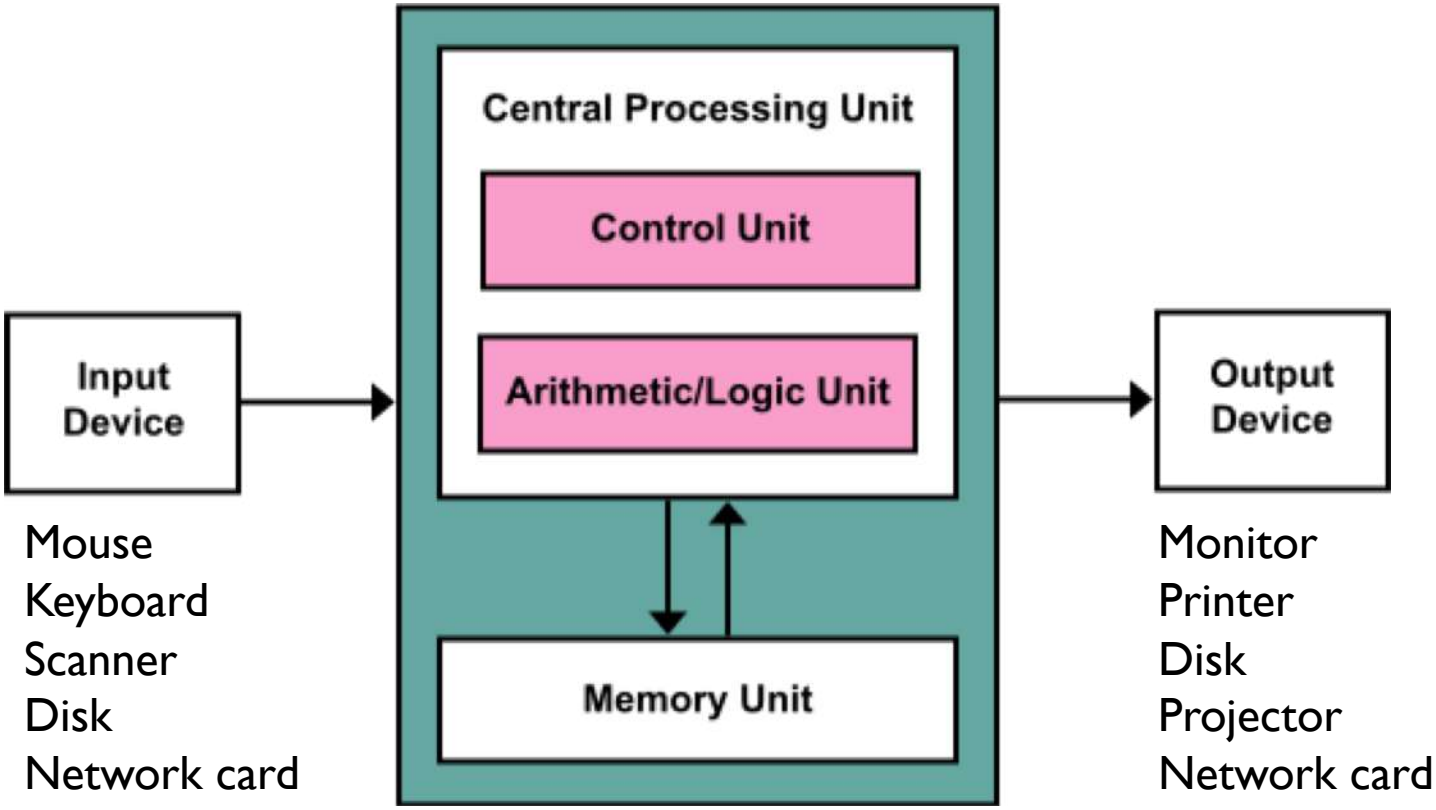
1.3 - Arquitectura

- Estructura básica
  - Una memoria que contiene instrucciones y datos
  - Una unidad de procesado para ejecutar operaciones aritméticas y lógicas
  - Una unidad de control para interpretar las instrucciones

- Comportamiento funcional
  - Máquina capaz de ejecutar instrucciones u órdenes elementales denominadas instrucciones máquina: instrucciones como acciones y decisiones.
  - La máquina debe ser controlada por un conjunto de instrucciones con un pequeño número de elementos centrales de proceso.
  - Tanto la información (datos) como el programa (instrucciones) deben almacenarse en el interior del ordenador en formato binario (con un alfabeto compuesto exclusivamente de ceros y unos).

1.3 – Arquitectura
Elementos básicos

- **Estructura básica**
  - Una memoria que contiene instrucciones y datos
  - Una unidad de procesado para ejecutar operaciones aritméticas y lógicas
  - Una unidad de control para interpretar las instrucciones

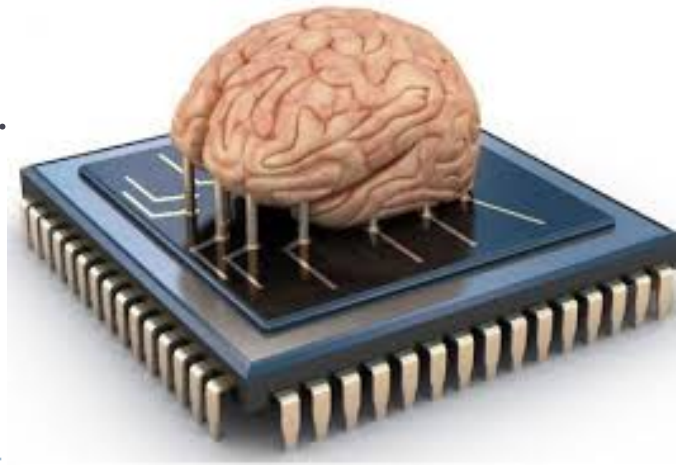


Mouse
Keyboard
Scanner
Disk
Network card
…

Monitor
Printer
Disk
Projector
Network card
…

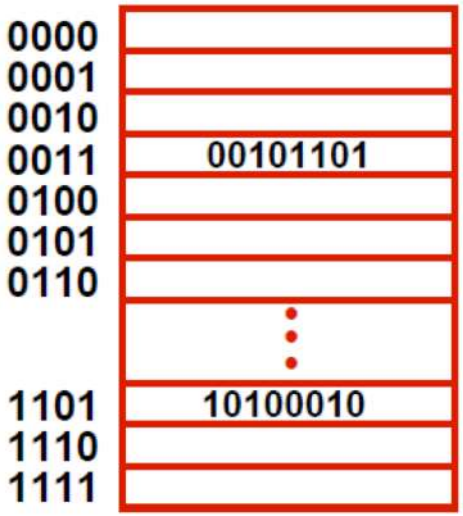
1.3 – Arquitectura
Central Processing Unit (CPU) o procesador

- Contiene el hardware necesario para procesar las instrucciones y los datos. La ejecución se produce siguiendo una secuencia de instrucción tras instrucción (a no ser que dicha secuencia se modifique explícitamente).
- Funciona sincronizado con un reloj. La capacidad de cálculo de la CPU depende de ese reloj y de la arquitectura interna.
- Unidad de control (CU)
  - Administra los recursos del computador, es decir, la memoria, los dispositivos de entrada, los de salida y los de almacenamiento.
  - Decodifica y ejecuta las instrucciones.
  - Realiza las transferencias de registros.
  - Transfiere los datos a la ALU, para su procesamiento.
- Unidad aritmético-lógica (ALU)
  - Ejecuta operaciones aritméticas entre operandos
  - Ejecuta pruebas lógicas entre operandos



1.3 – Arquitectura
Memoria

- Almacena datos mediante lectura y escritura sobre una posición de la memoria indicada mediante una dirección
- Almacena programas (instrucciones)



- Tiene capacidad para un cierto número de bits organizados en palabras de memoria
  - Matriz de 2^k x m bits
  - Dirección: identificador único (k bits) de localización en la memoria
  - Contenido: valor de m bits almacenado en la dirección

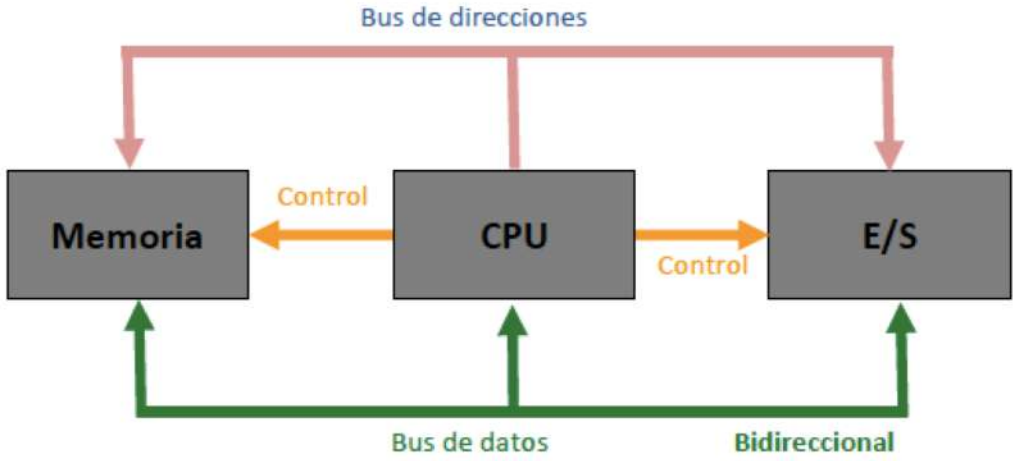
- En un computador suele haber varios tipos de memoria (jerarquía de memorias)
  - Registros de la CPU, memoria caché, memoria principal, memoria secundaria (de más rápida, más cara y menor capacidad a lo contrario)

1.3 – Arquitectura
Dispositivos de entrada y salida (periféricos)

- Realizan la transferencia de información con los dispositivos periféricos (disco duro, terminales, impresoras, etc.)
- Tipos
  - Entrada (E): recibe nuevos datos e instrucciones
  - Salida (S): entrega resultado en algún formato
  - Algunos periféricos pueden ser de E/S, por ejemplo la red, cd/dvd regrabables, etc.
- Los programas que controlan el acceso a los periféricos se llaman generalmente driver (controlador)
- Puntos a tener en cuenta:
  - Los periféricos pueden ser dispositivos electromecánicos y su manera de operar es diferente de la del procesador y la memoria que son dispositivos electrónicos.
  - Los periféricos suelen tener una velocidad muy inferior al procesador.
  - La operación de los periféricos se tiene que sincroniza con la del procesador y la memoria, y además se tiene que controlar para que no perturbe la operación del procesador central y la de otros periféricos conectados al sistema.

1.3 – Arquitectura
Buses

- Permiten la transferencia de información entre todos los bloques que constituyen el ordenador
- Tipos de buses según funcionalidad
  - **Bus de dirección**: conjunto de líneas o canales que utiliza el procesador para indicar la dirección de memoria donde se encuentra el dato o la instrucción a la que se quiere acceder.
  - **Bus de datos**: canales por donde se transfiere el contenido de la posición de memoria seleccionada. Es un bus bi-direccional.
  - **Bus de control**: conjunto de líneas que controlan la transferencia a realizar. Indican si se realiza una operación de lectura, escritura, acceso a memoria o periférico, reloj, etc.



1.3 – Arquitectura
Ejecución de un programa

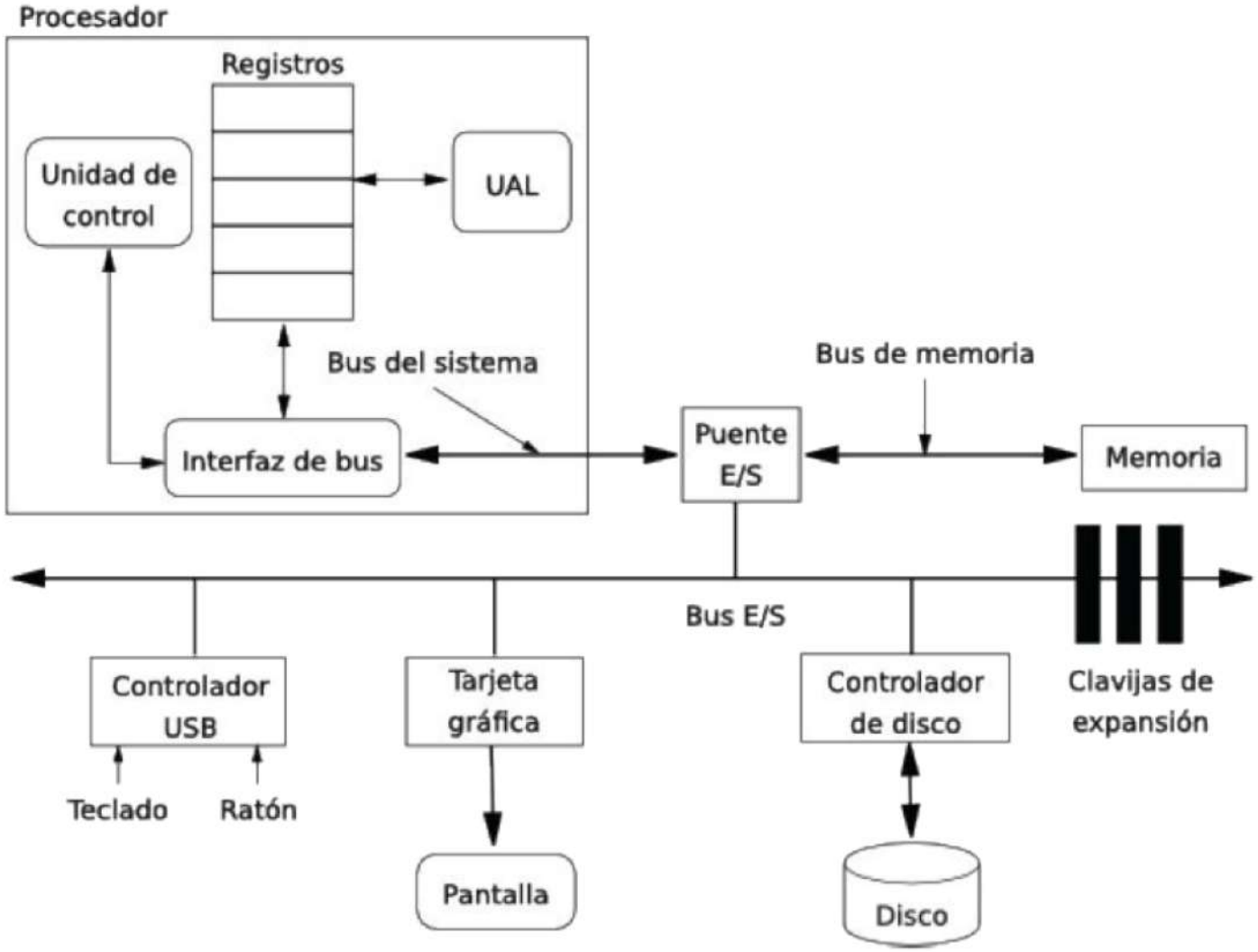
- Un programa es una sucesión de órdenes que la CPU va leyendo de la memoria y ejecutando de forma consecutiva.
- Proceso básico de ejecución de un programa en la máquina de Von Neumann:

1. Se lleva una instrucción desde la memoria a la unidad de control (y el PC se mueve a la siguiente línea)
2. La unidad de control interpreta la instrucción y la ejecuta
   - Buscar información de la memoria (si hace falta) y la lleva a los registros de la ALU
   - Ejecutar la instrucción (sumas, restar, mirar entrada, escribir en salida,...)
   - Poner la información resultante en la memoria (si es necesario)
3. Mientras no se llegue al final del programa volver al paso (1)

1.3 – Arquitectura
Ejecución de un programa

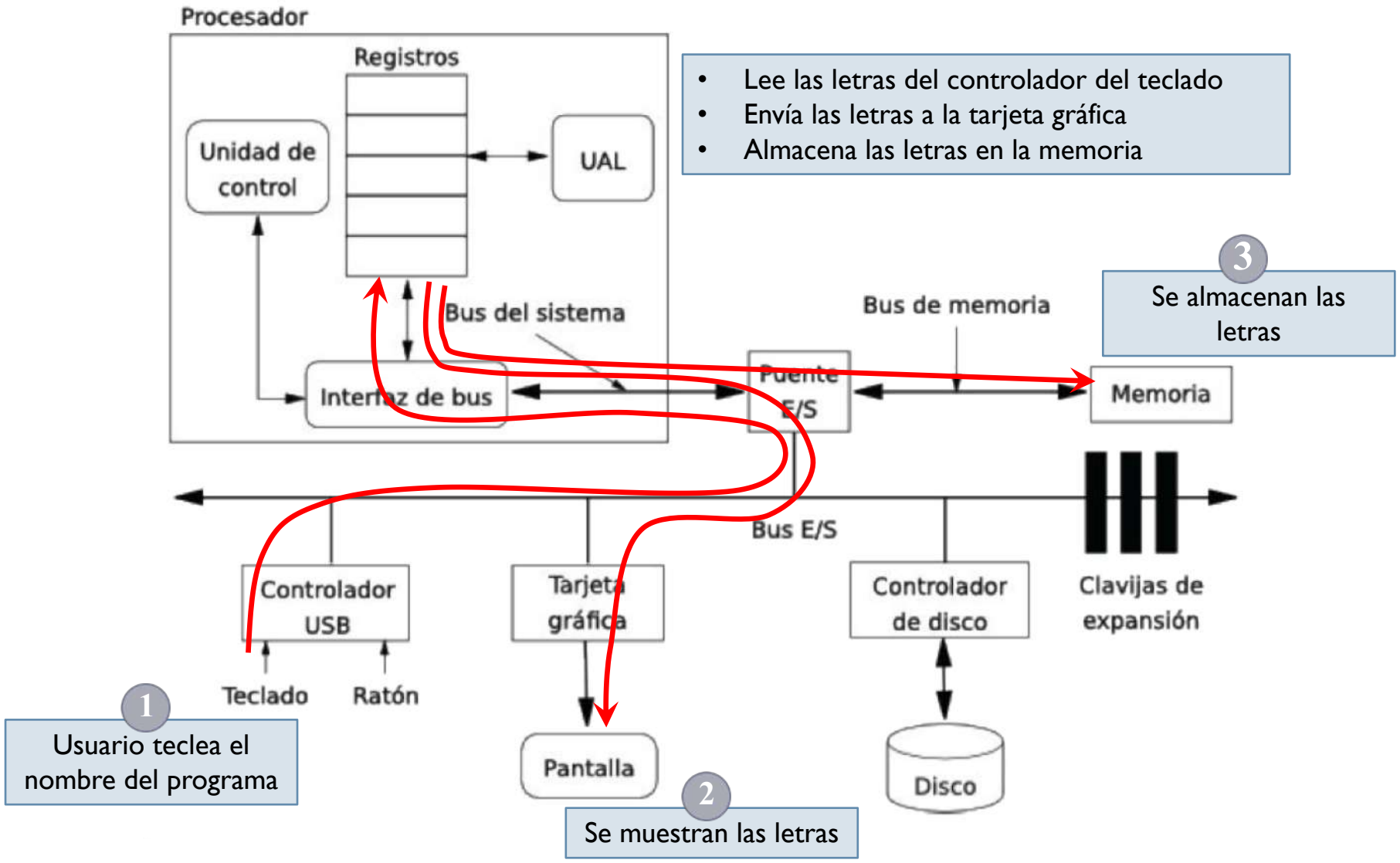
- Supongamos que se abre un terminal y se ejecuta el programa "HelloWorld"
- Los pasos para su ejecución son
  1) Usuario teclea el nombre del programa
  2) Se obtiene el programa del disco duro y se almacena en la memoria
  3) El procesador ejecuta una tras otras las instrucciones del programa
  4) Se borra el programa y sus datos de la memoria

1.3 – Arquitectura
Ejecución de un programa



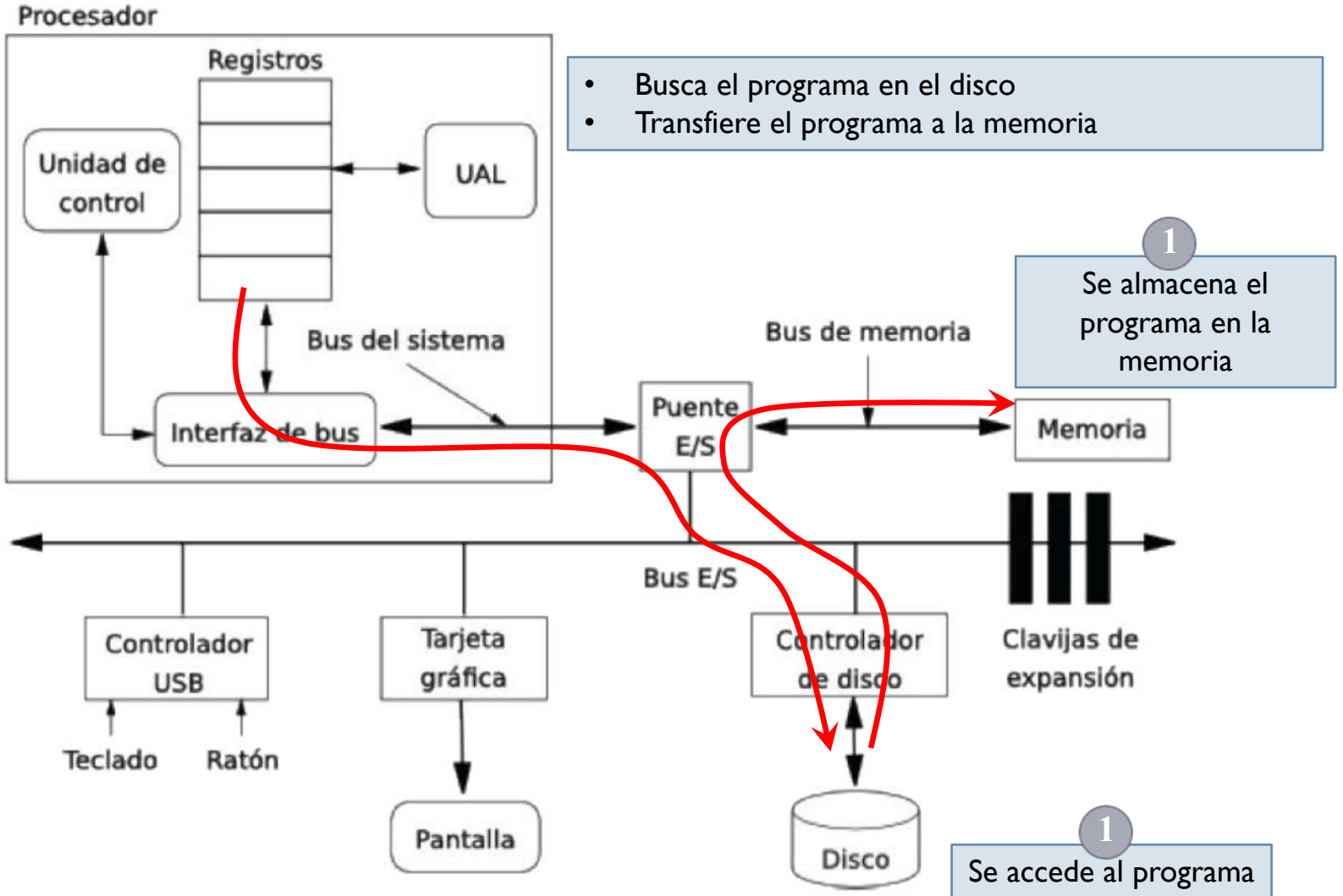
1.3 – Arquitectura

Ejecución de un programa: paso 1



1.3 – Arquitectura

Ejecución de un programa: paso 2



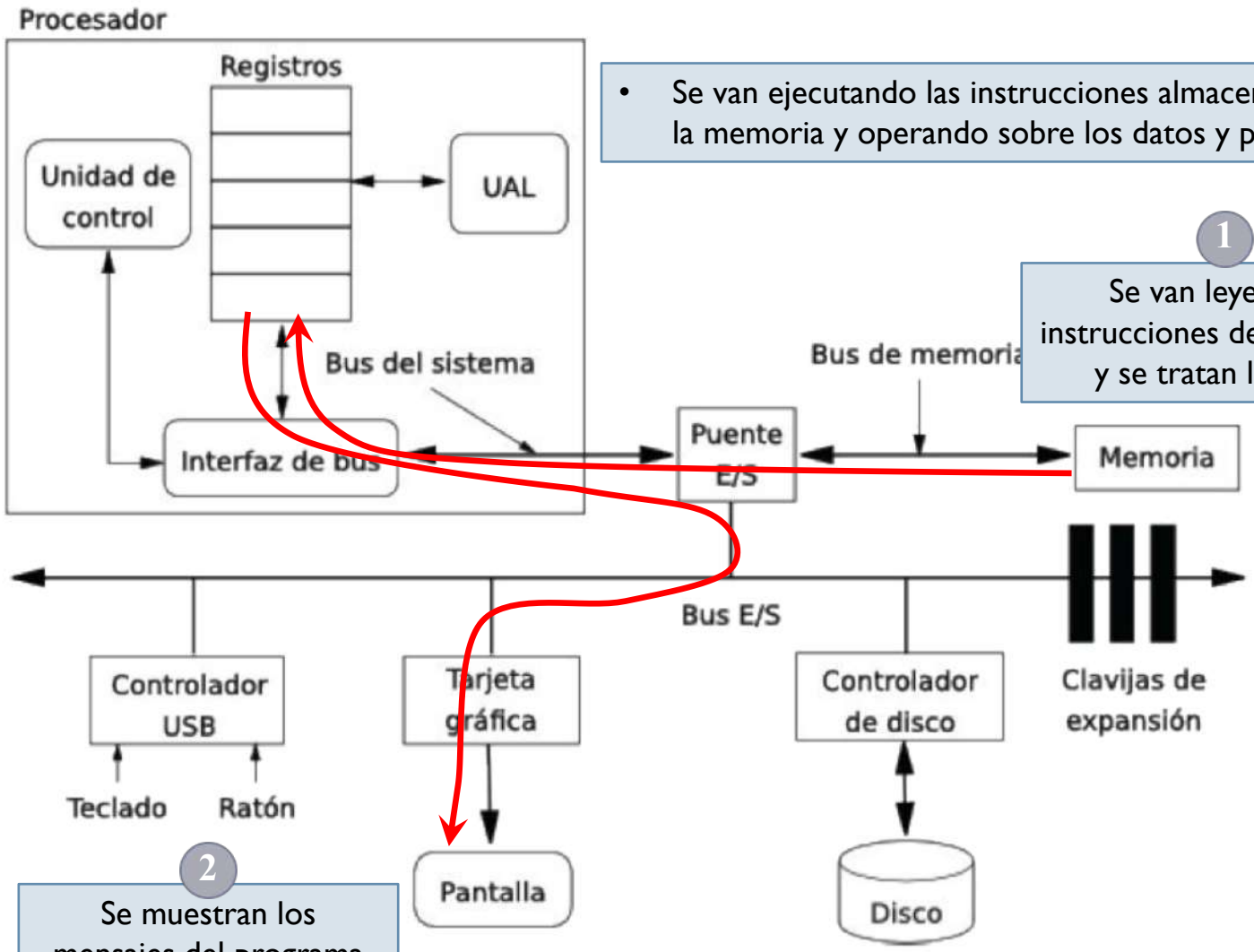
- Busca el programa en el disco
- Transfiere el programa a la memoria

1 Se almacena el programa en la memoria

1 Se accede al programa (fichero almacenado en el disco duro)

1.3 – Arquitectura

Ejecución de un programa: paso 3



- Se van ejecutando las instrucciones almacenada en la memoria y operando sobre los datos y periféricos

1

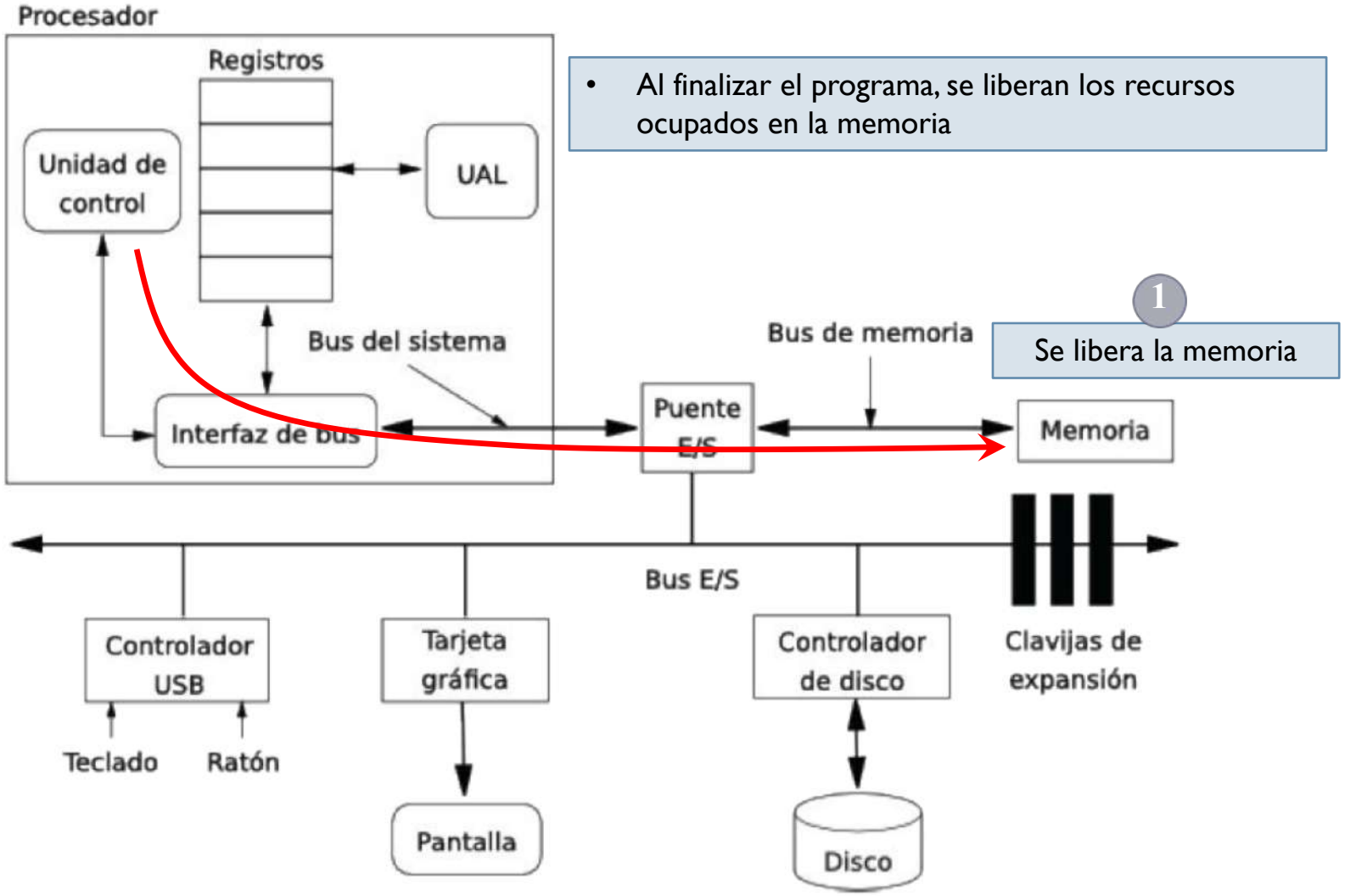
Se van leyendo las instrucciones de la memoria y se tratan los datos

2

Se muestran los mensajes del programa

1.3 – Arquitectura

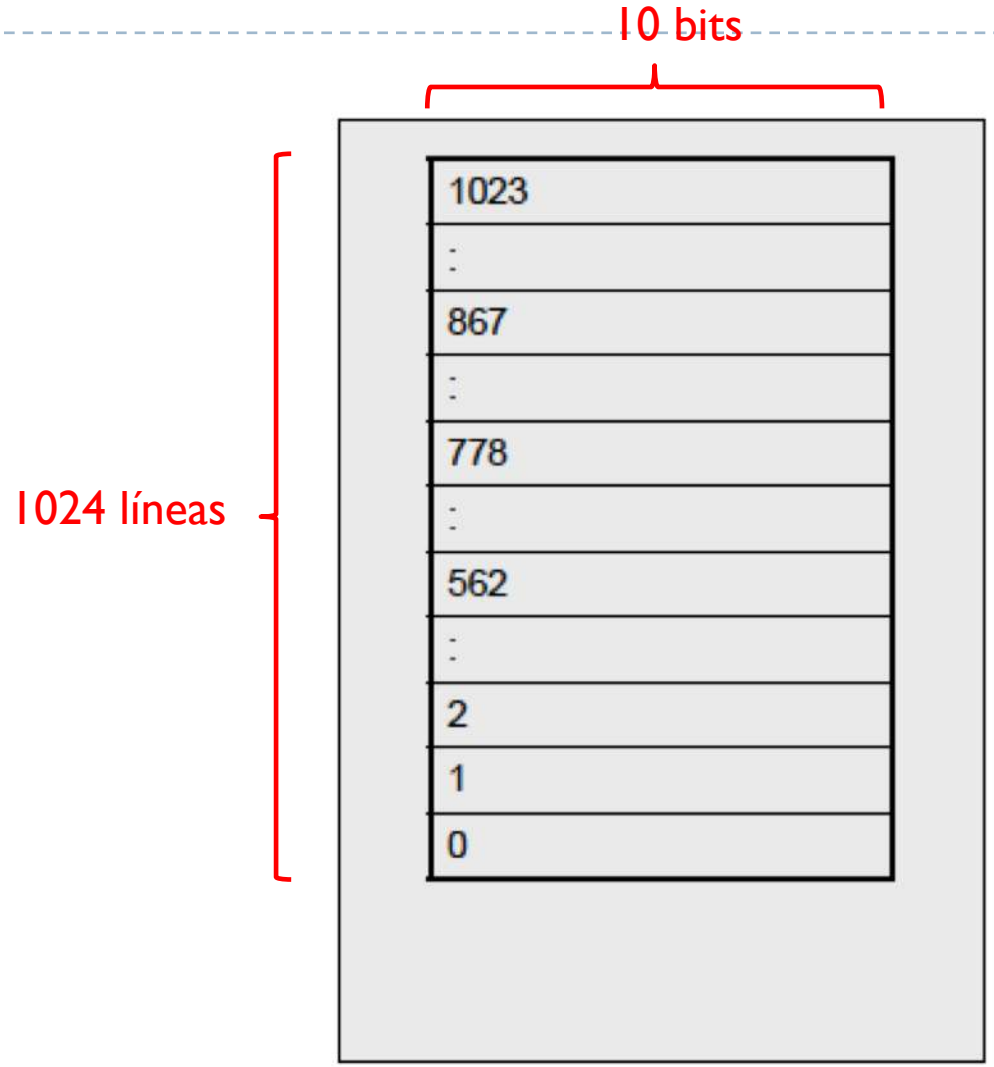
Ejecución de un programa: paso 4



- Al finalizar el programa, se liberan los recursos ocupados en la memoria

1.3 – Arquitectura

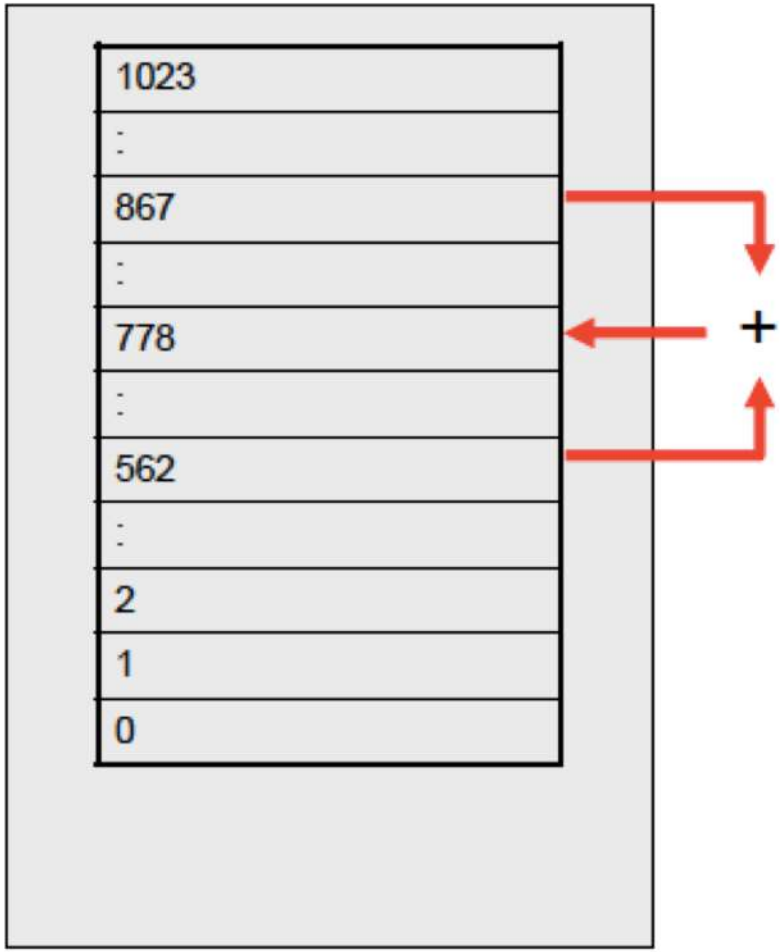
Ejemplo



Memoria de 1024 x 10 bits = 10240 bits = 1280 bytes = 1,25 kbytes

1.3 – Arquitectura
Ejemplo

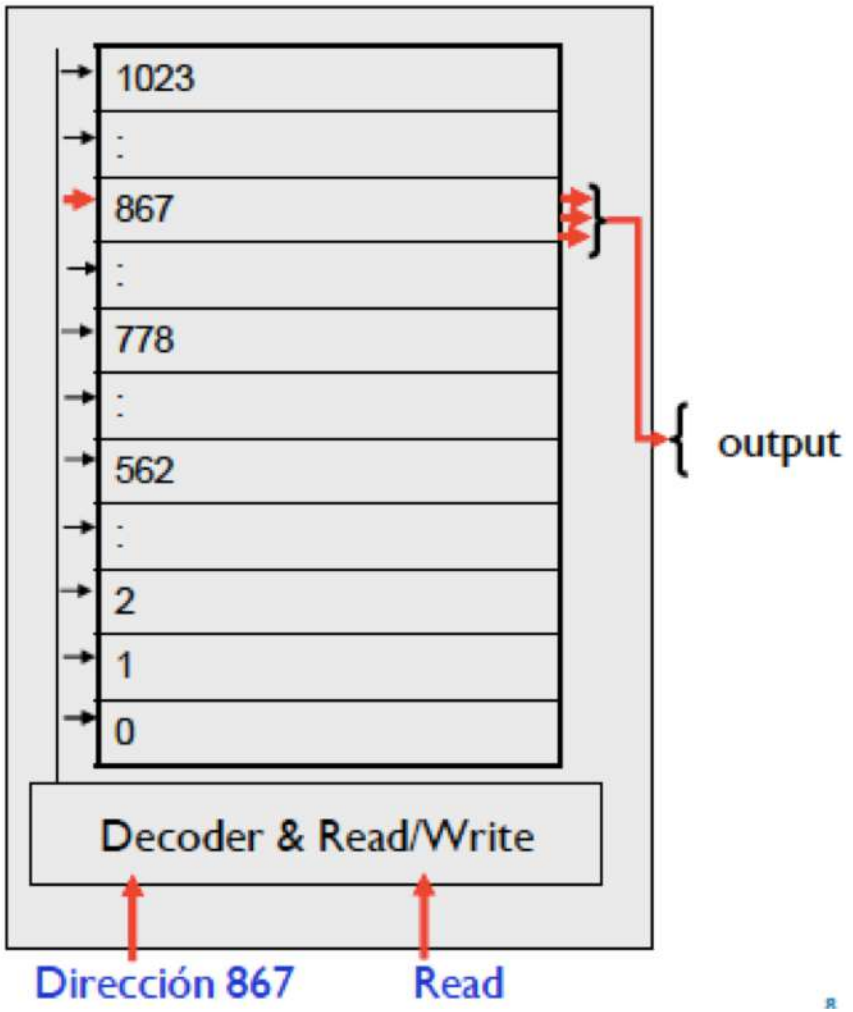
- Operación
  - Sumar el contenido de la dirección 867 con el contenido de la dirección 562 y almacenar el resultado en la posición 778



Memoria de 1024 x 10 bits = 1,25 kbytes

1.3 – Arquitectura
Ejemplo

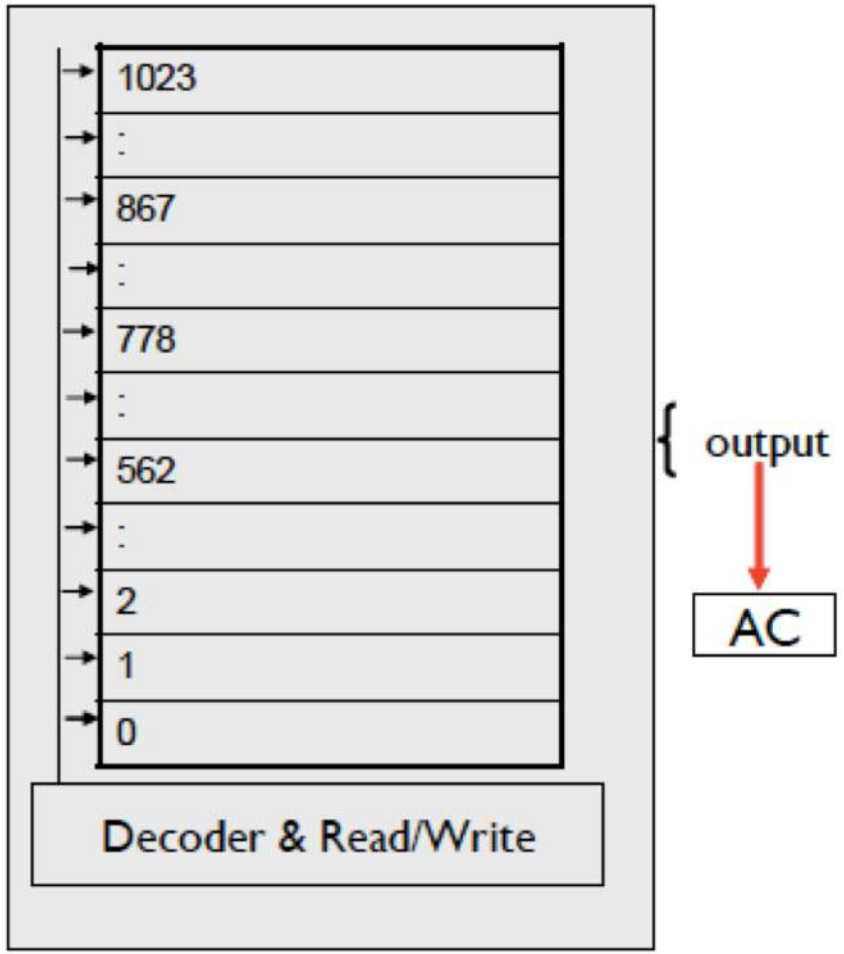
- Solución
  - Leer la memoria en la dirección 867



Memoria de 1024 x 10 bits = 1,25 kbytes

1.3 – Arquitectura
Ejemplo

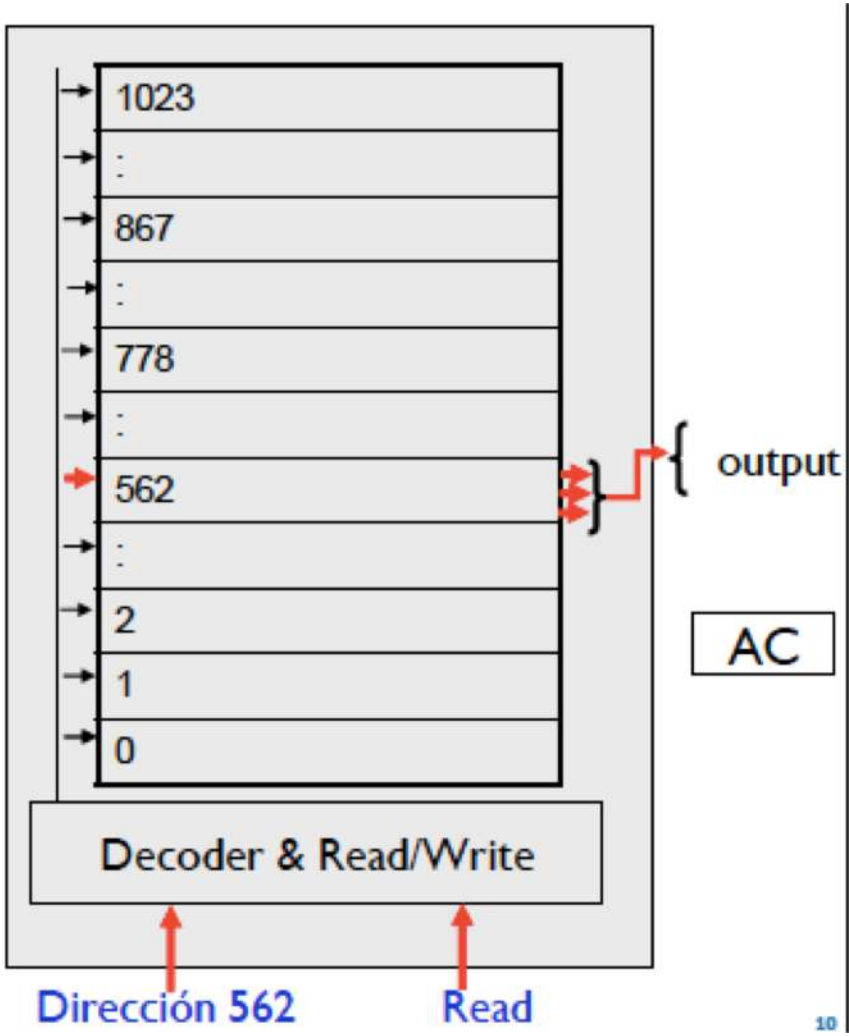
- Solución
  - Leer la memoria en la dirección 867
  - Almacenar el contenido en un registro externo a la memoria y propio de la CPU (en el ejemplo llamado AC)



Memoria de 1024 x 10 bits = 1,25 kbytes

1.3 – Arquitectura
Ejemplo

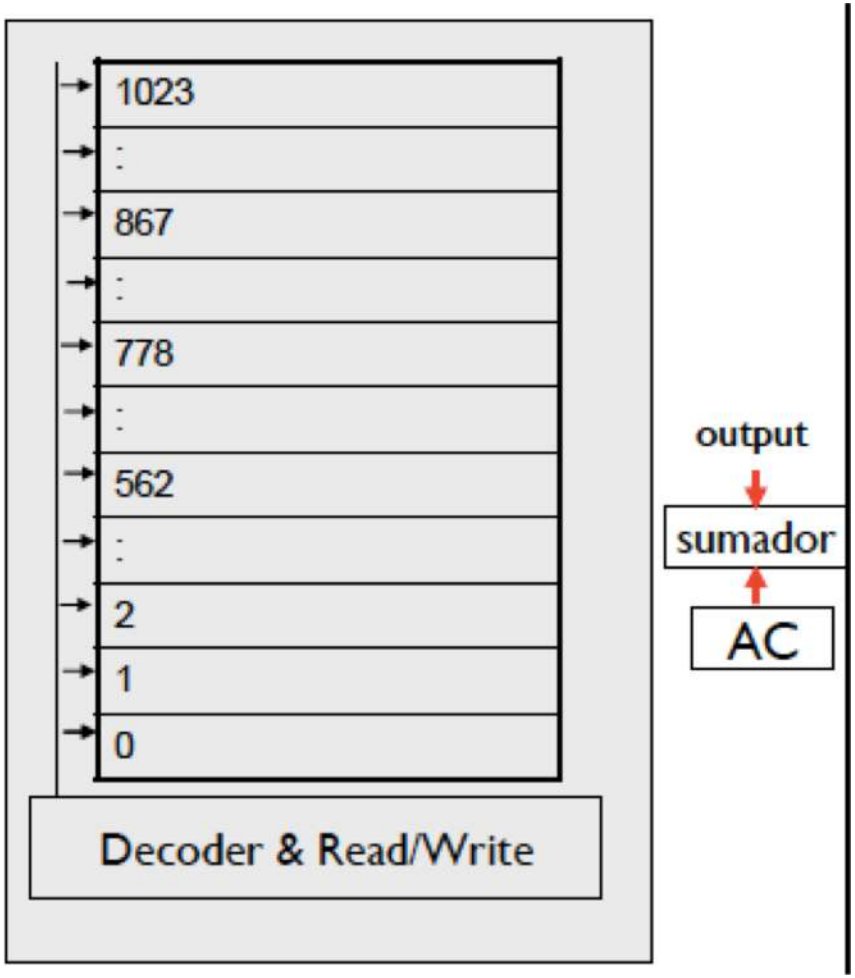
- Solución
  - Leer la memoria en la dirección 867
  - Almacenar el contenido en un registro externo a la memoria y propio de la CPU (en el ejemplo llamado AC)
  - Leer la memoria en la dirección 562



Memoria de 1024 x 10 bits = 1,25 kbytes

1.3 – Arquitectura
Ejemplo

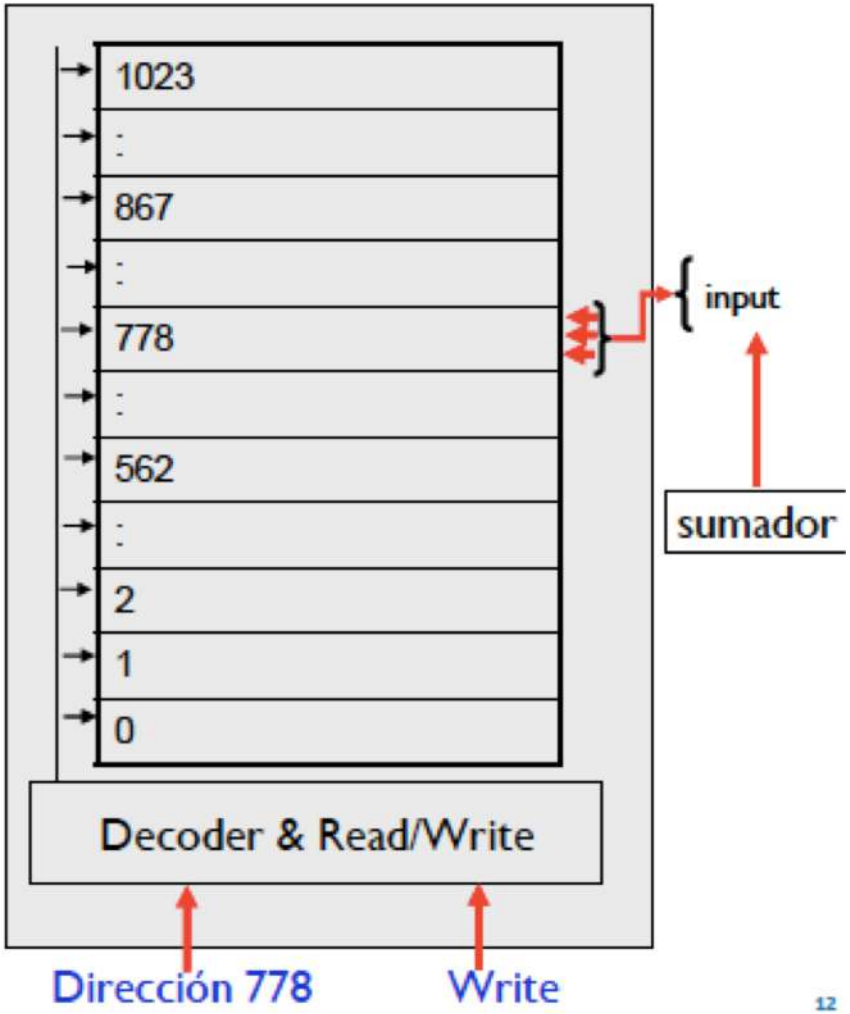
- Solución
  - Leer la memoria en la dirección 867
  - Almacenar el contenido en un registro externo a la memoria y propio de la CPU (en el ejemplo llamado AC)
  - Leer la memoria en la dirección 562
  - Sumar el contenido con el contenido del registro AC



Memoria de 1024 x 10 bits = 1,25 kbytes

1.3 – Arquitectura
Ejemplo

- Solución
  - Leer la memoria en la dirección 867
  - Almacenar el contenido en un registro externo a la memoria y propio de la CPU (en el ejemplo llamado AC)
  - Leer la memoria en la dirección 562
  - Sumar el contenido con el contenido del registro AC
  - Almacenar el contenido del registro externo en la posición 778



Memoria de 1024 x 10 bits = 1,25 kbytes

1.3 – Arquitectura
Registros

- Almacenamiento temporal
- El número y función de los registros varía entre los diferentes diseños del procesador
- Es una de las decisiones más importantes en el diseño del procesador
- Se encuentran en el nivel más alto de la jerarquía de memoria (más rápida, más cara, menor capacidad)
- Hay visibles y de control/estado

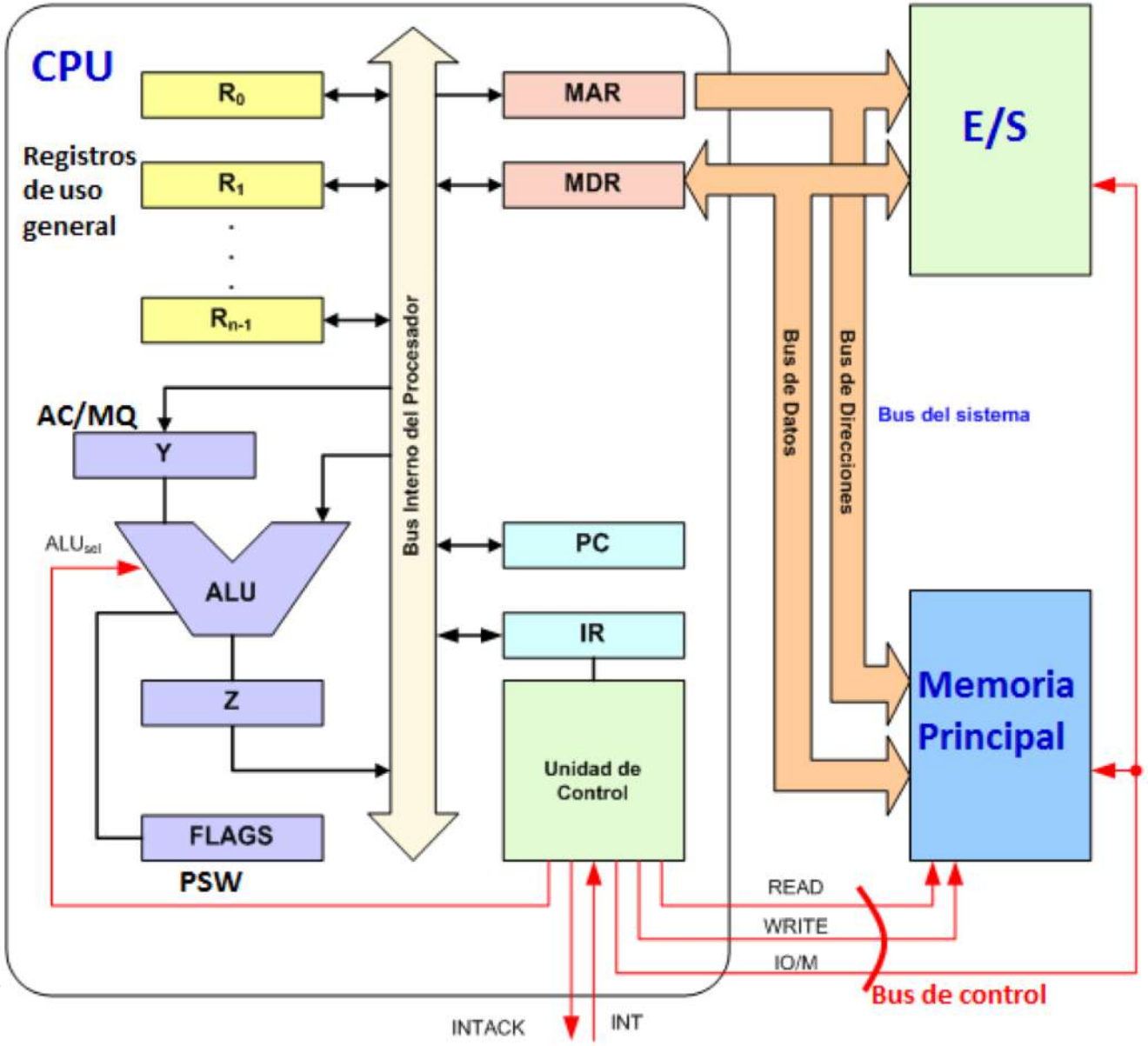
1.3 – Arquitectura
Registros visibles

- Pueden ser de
  - Uso general
  - Datos (operandos del ALU)
  - Direcciones
  - Código de condición (después de una operación contienen resultados implícitos que se pueden usar para saltos condicionales, p.e. salta si el resultado fue 0)
- Ejemplos
  - AC (Accumulator) y MQ (Multiplier Quotient) son registros que se emplean en las operaciones de la ALU

1.3 – Arquitectura
Registros de control/estado

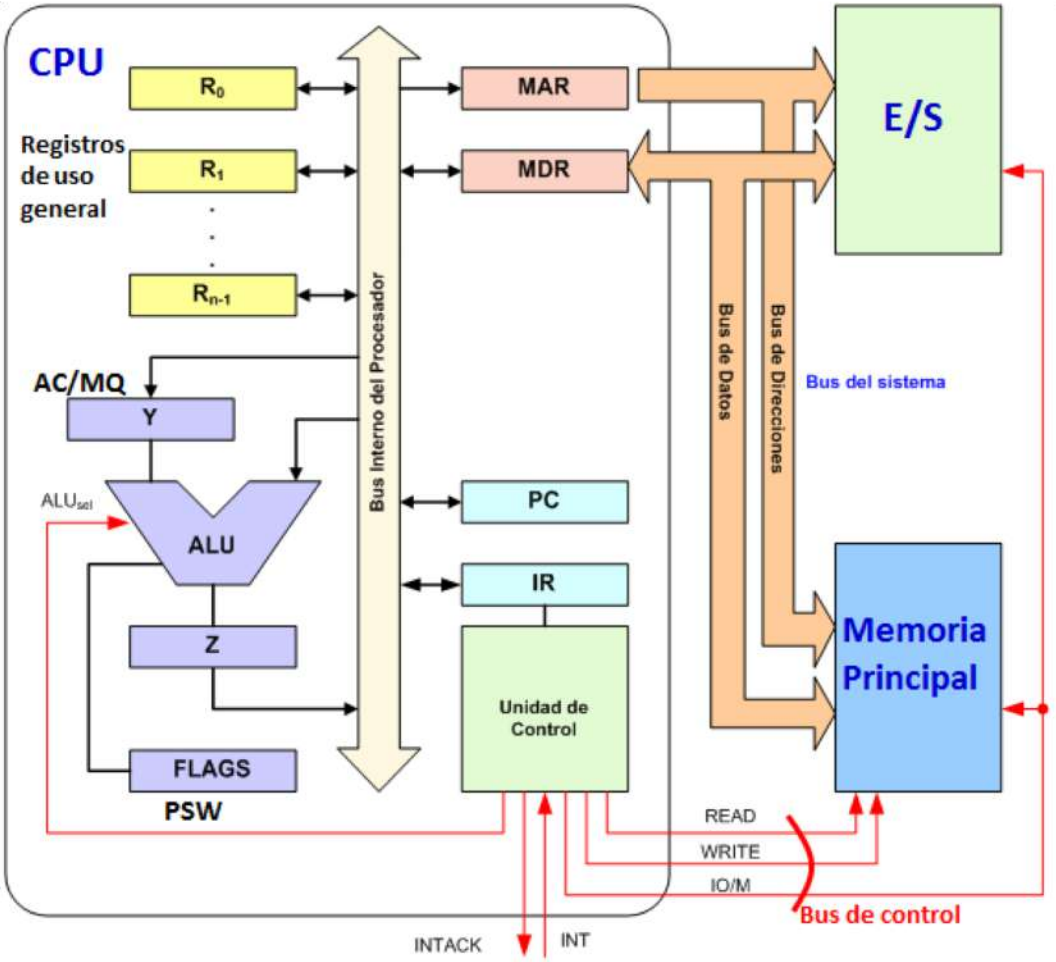
- Se usan para controlar el funcionamiento del procesador
- Ejemplos
  - PC (Program Counter). Contiene la dirección en la memoria de la siguiente instrucción a ser ejecutada.
  - IR (Instruction Register). Contiene la última instrucción captada. Contiene el código de operación de la instrucción que se va a ejecutar.
  - MAR (Memory Address Register). Contiene la dirección de una posición de memoria. Especifica la dirección en memoria de la palabra que va a ser escrita o leída en MDR.
  - MDR (Memory Data Register). Contiene una palabra de datos que debe ser almacenada (escribir) en la memoria, o es usado para recibir (leer) una palabra procedente de la memoria

1.3 – Arquitectura



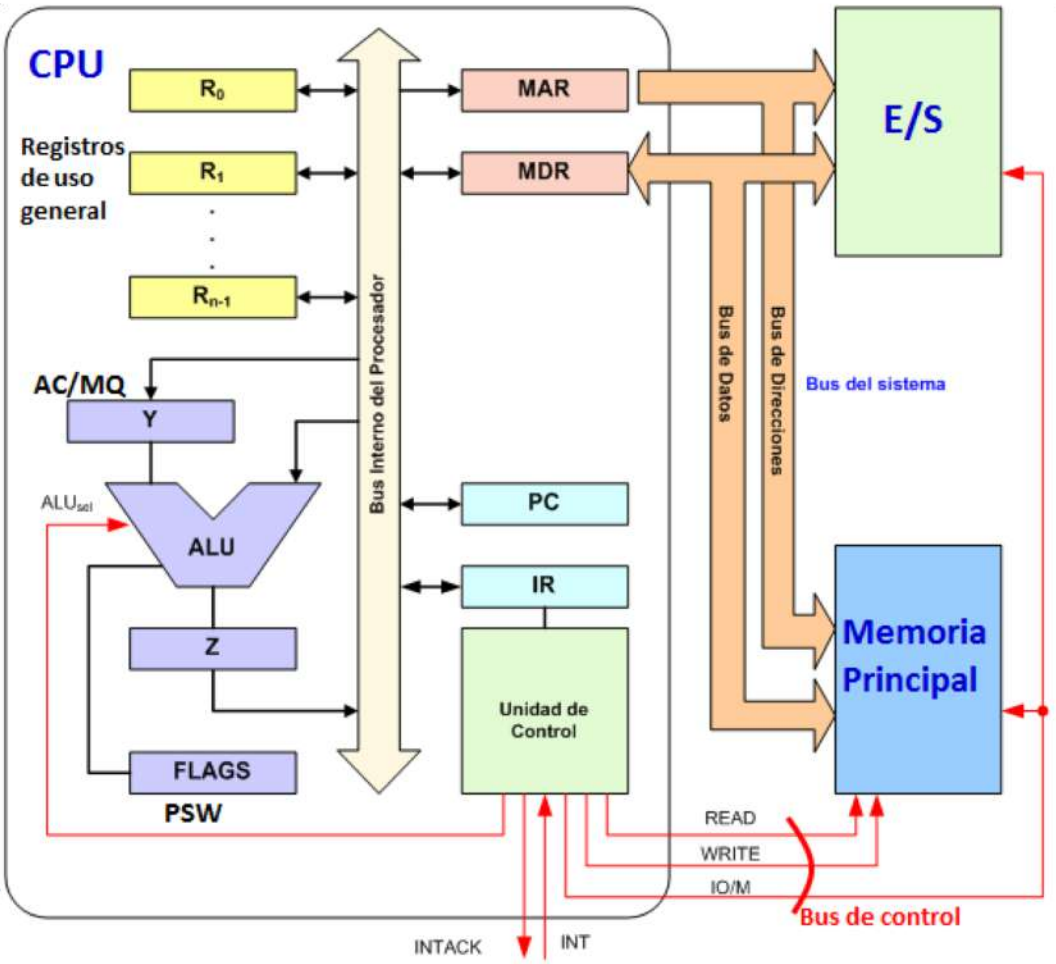
1.3 – Arquitectura
Paso de valores CPU - Memoria

- Interfaz Unidad de Control (CU) y memoria
  - Esta capta un valor desde y hacia la memoria a través de los registros MAR y MDR
  - Para leer en la dirección @:
    - CU escribe la dirección @ en el registro MAR
    - CU envía la señal de lectura a la memoria
    - La memoria coge el valor en la dirección indicada en MAR y lo guarda el contenido en el registro MDR
    - CU lee el dato del registro MDR



1.3 – Arquitectura
Paso de valores CPU - Memoria

- Interfaz Unidad de Control (CU) y memoria
  - Esta capta un valor desde y hacia la memoria a través de los registros MAR y MDR
  - Para escribir en la dirección @:
    - CU escribe el valor X en el registro MDR
    - CU escribe la dirección @ en el registro MAR
    - CU envía la señal de escritura a la memoria
    - La memoria coge el valor contenido en MDR y lo guarda en la dirección indicada por MDR



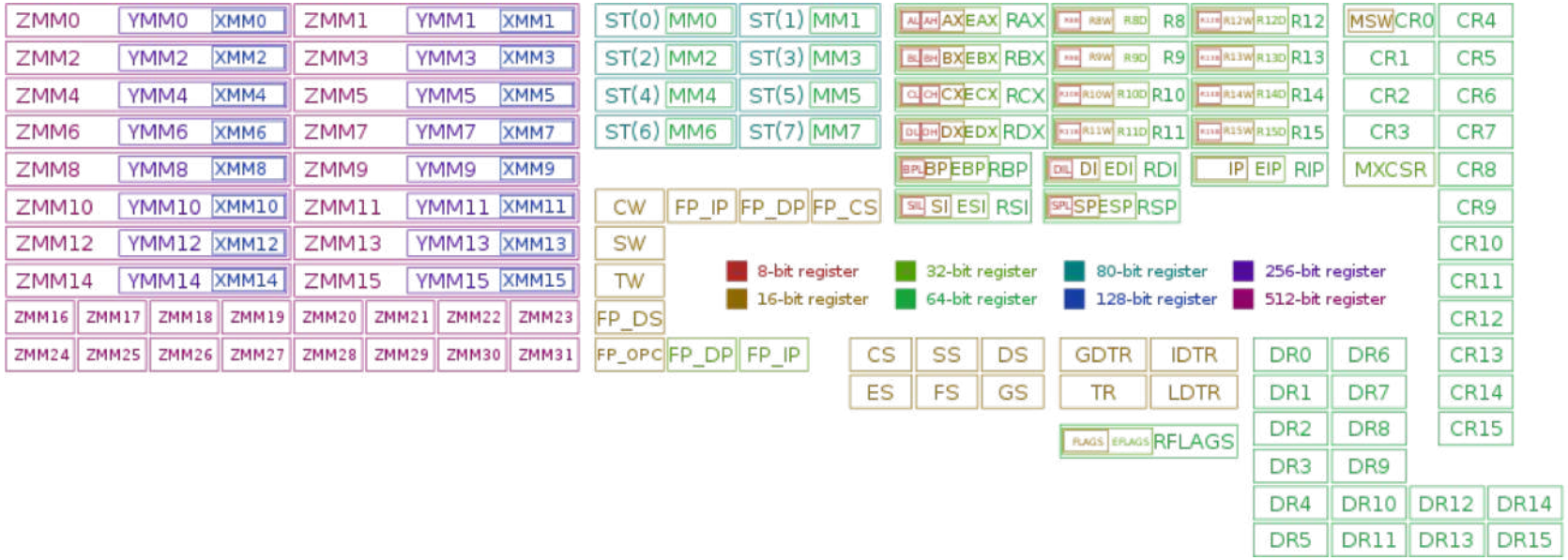
1.3 – Arquitectura
Evolución de los registros en los años

- Arquitectura x86
  - Intel 8086/8088: 14 registros de 16 bits
  - Intel 80386: 9 registros de 32 bits y 7 de 16 bits
  - Intel Pentium: 80386 + 8 registros de 64 bits
  - Intel Pentium III: Pentium + 1 registro de 32 bits + 8 registros de 128 bits
- Arquitectura x86-64
  - AMD Opteron: todos pasan a 64 bits + 8 registros de 64 bits

1.3 – Arquitectura

Evolución de los registros en los años

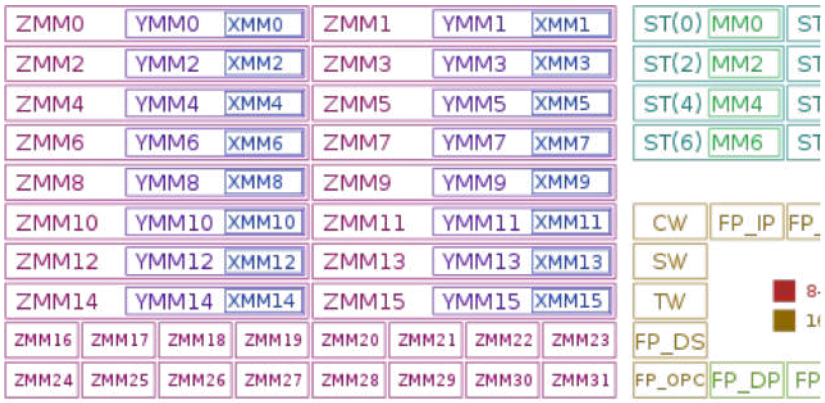
- ¿Y ahora cuantos hay?

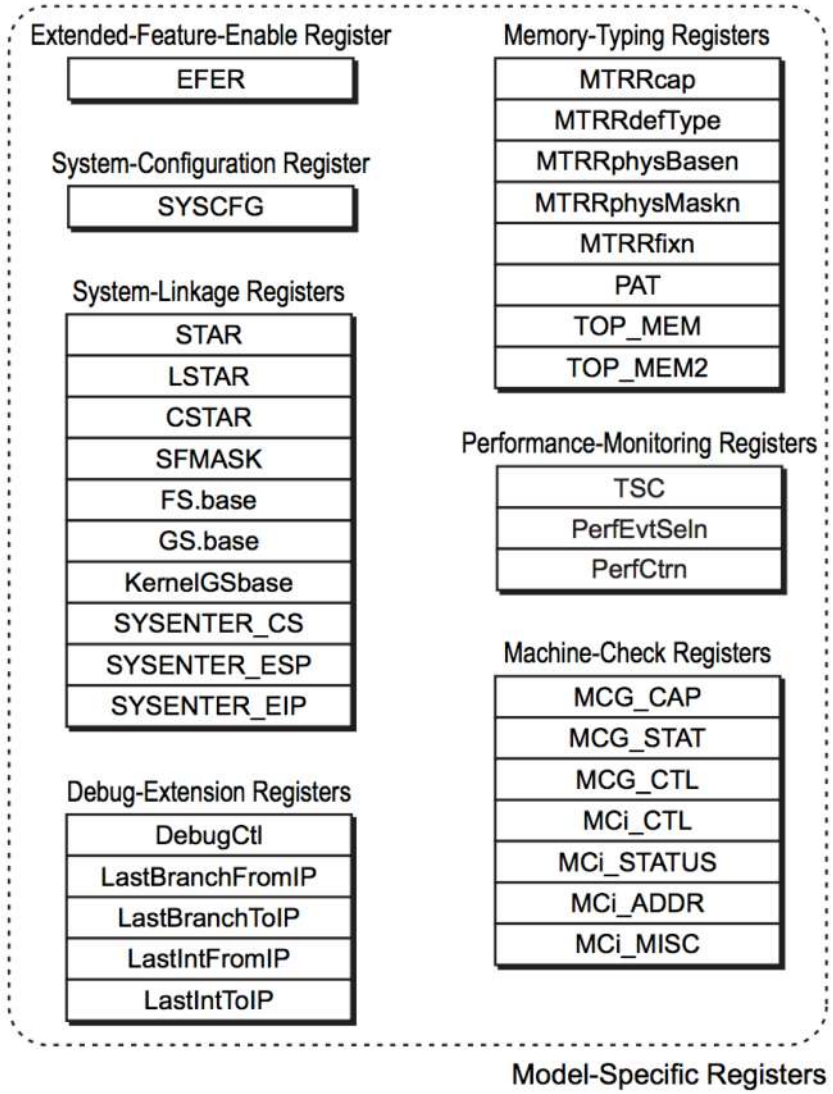


1.3 – Arquitectura

Evolución de los registros en los años

- ¿Y ahora cuantos hay?





Reference: AMD64 Architecture Manual

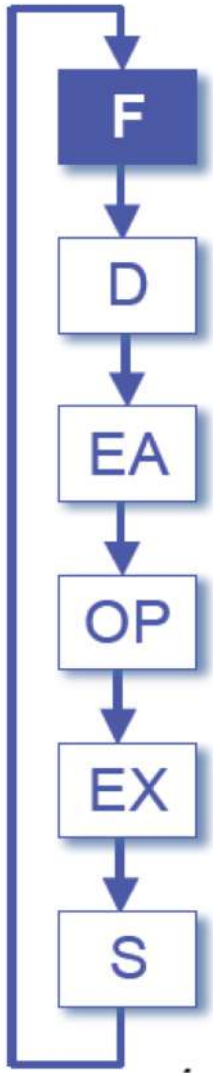
1.3 – Arquitectura
La instrucción

- La instrucción es la unidad fundamental
- Especifica dos cosas
  - Opcode: el código de la operación a ejecutarse
  - Operands: el dato y la dirección que hay que usar en la operación
- Una instrucción es codificada como secuencia de bits
  - Cada código significa una operación concreta
  - Este código depende de la arquitectura y puede ser de longitud fija o variable (1, 2 o 4 bytes)
    - Por ejemplo en la arquitectura x86, el código 00000000 significa hacer una adición desde un registro a la memoria usando un operando de 8 bits
  - La unidad de control sabe interpretar estos códigos y enviar las señales a los diferentes componentes en un orden concreto para ejecutar la operación
- Las instrucciones posibles y sus formatos se conocen como Instruction Set Architecture (ISA)

1.3 – Arquitectura

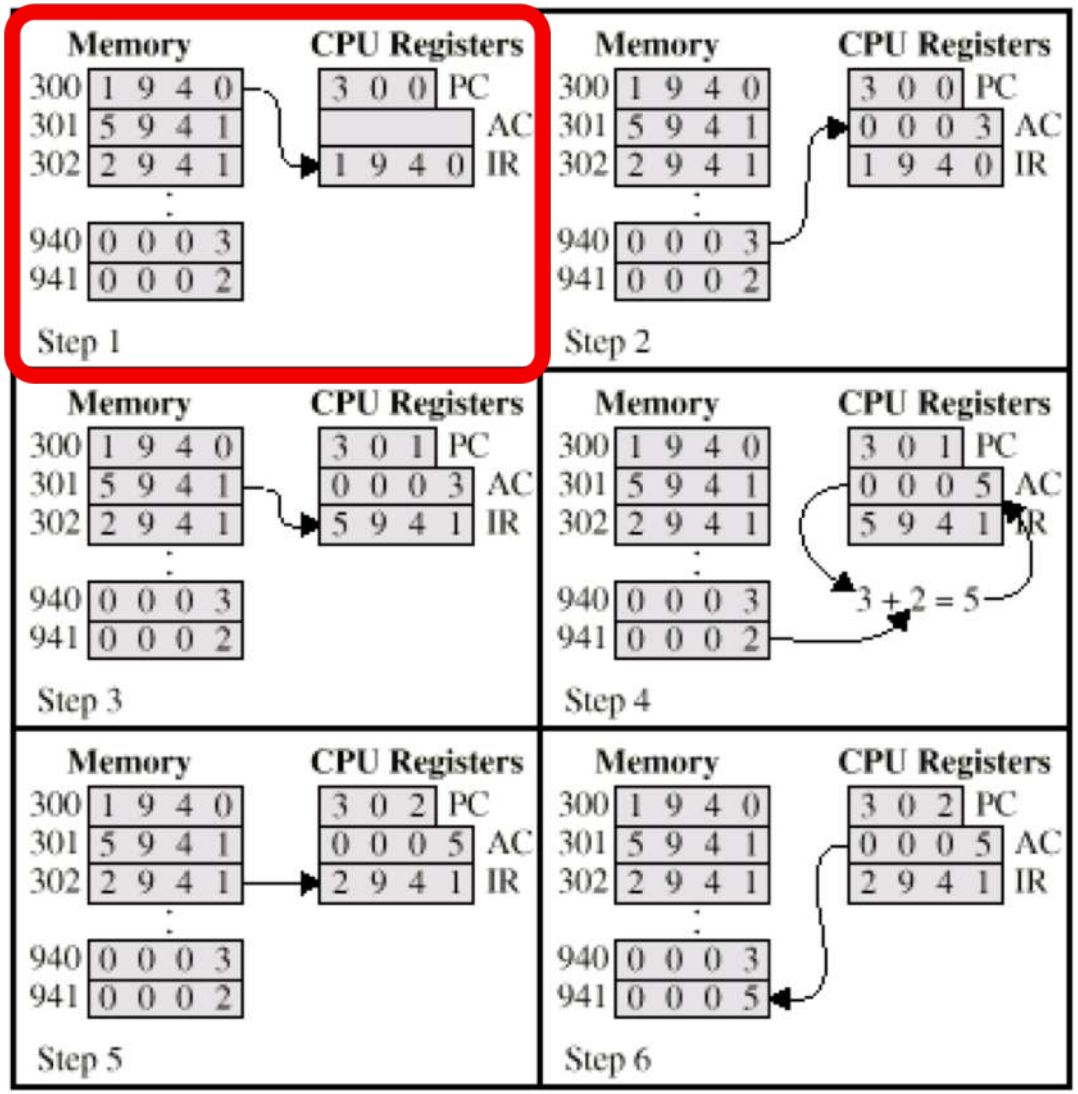
La instrucción

- **Captar instrucción** o **Fetch de instrucción**: el procesador lee una instrucción de la memoria. Las instrucciones y los datos deben ser cargados desde la memoria a los registros de la CPU.
- **Interpretar instrucción**: la instrucción se decodifica para determinar qué acción es necesaria.
- **Evaluar la direcciones**: para instrucciones que requieren acceder a memoria, calcular la dirección (diferentes tipos de direccionamiento).
- **Captar datos o Fetch de datos**: la ejecución de una instrucción puede exigir leer datos de la memoria o de un módulo I/O.
- **Procesar datos**: en la ejecución se puede exigir llevar a cabo alguna operación aritmética o lógica con los datos.
- **Escribir datos**: los resultados de la ejecución pueden exigir escribir datos en la memoria o en un módulo I/O.
- Cada instrucción no ejecuta necesariamente las 6 etapas
  Depende de la instrucción misma



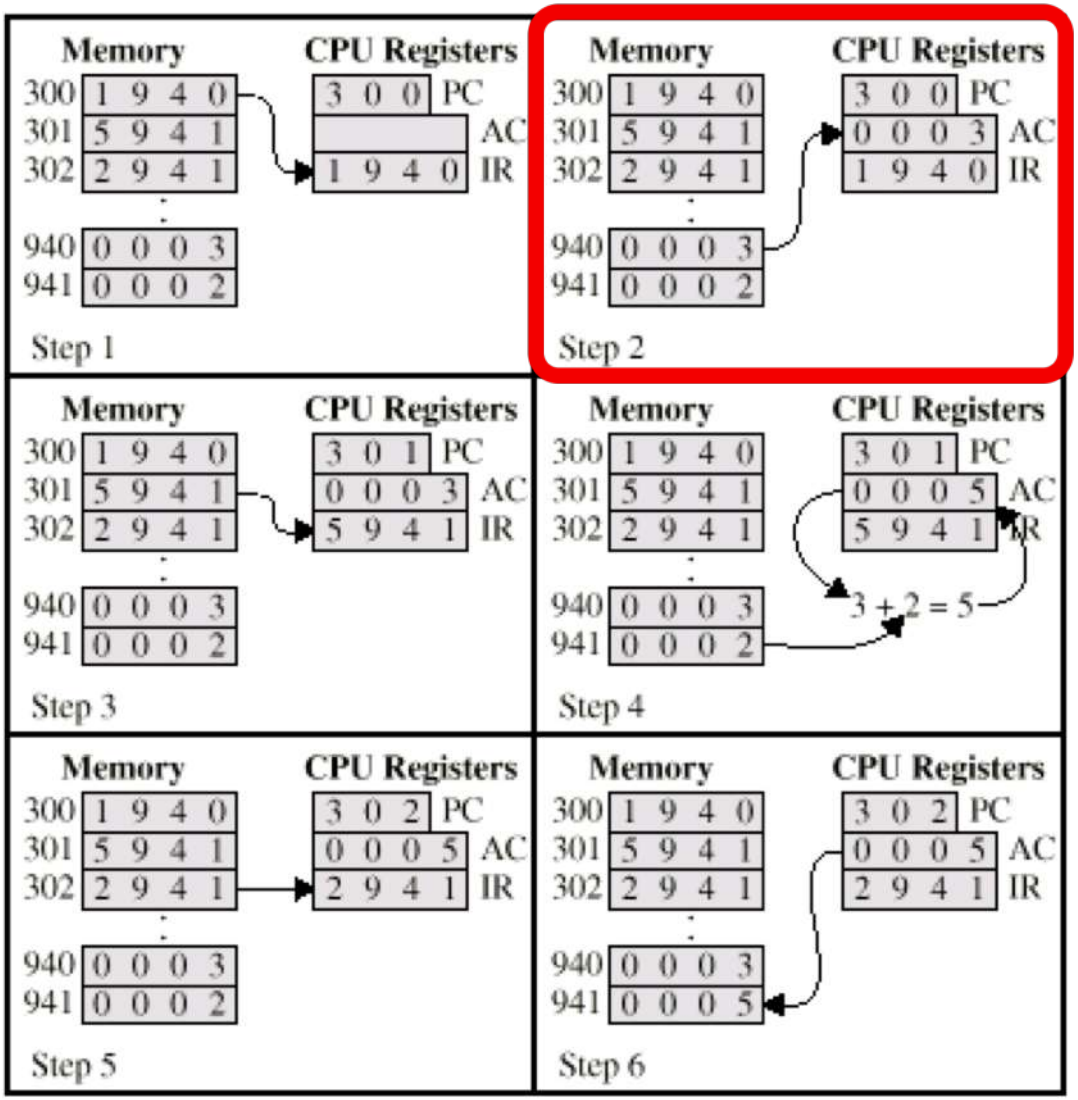
1.3 – Arquitectura
Funcionamiento

- Ejemplo completo
  1) PC contiene el valor 300
     Se carga esta instrucción en IR, usando MDR y MAR



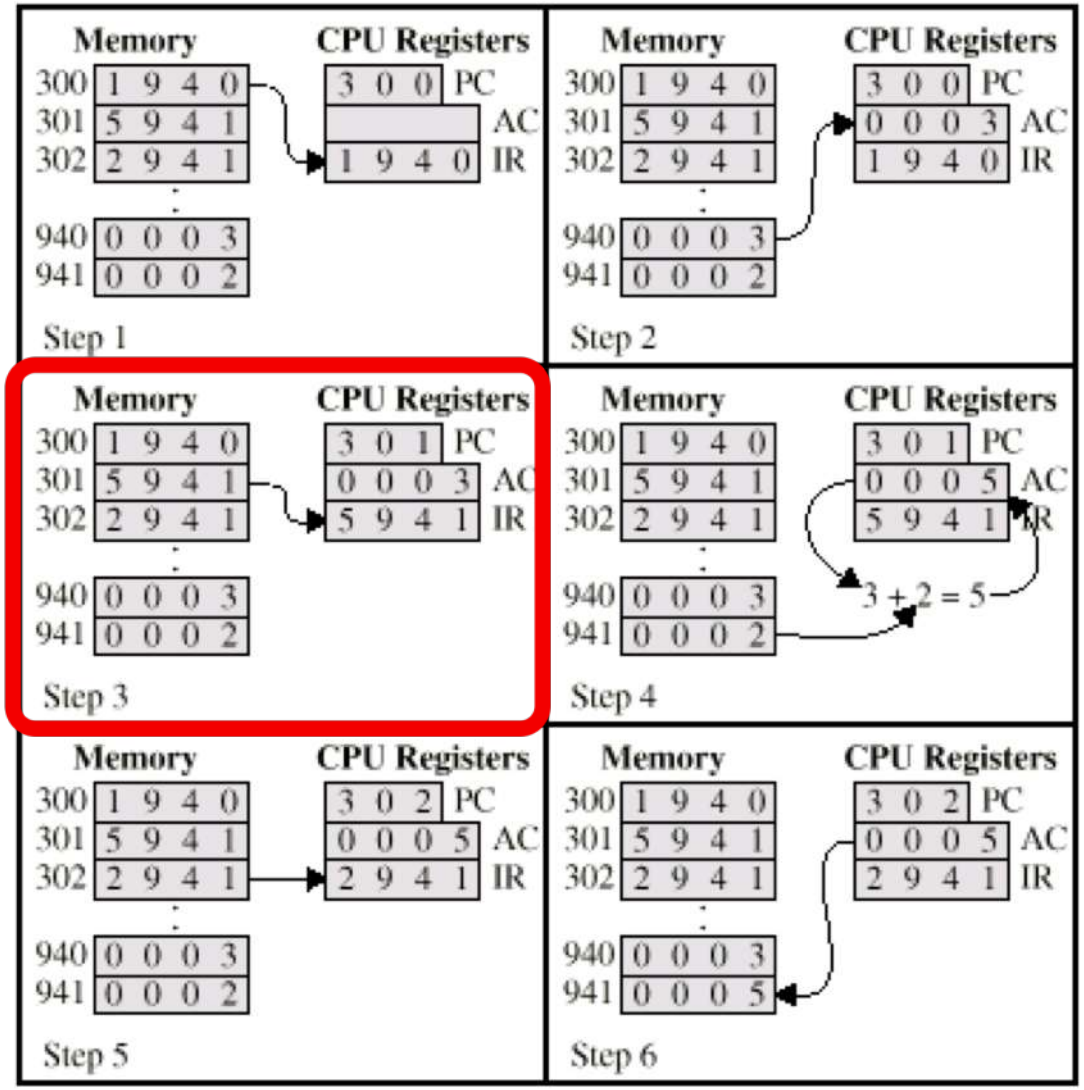
1.3 – Arquitectura
Funcionamiento

- Ejemplo completo
  2) Los primeros 4 bits de IR (el 1 decimal → 0001 binario) indican la instrucción
     En este caso indica que hay que cargar en el registro AC el contenido de la dirección 940 (es decir 1001 0100 0000 en binario)



1.3 – Arquitectura
Funcionamiento

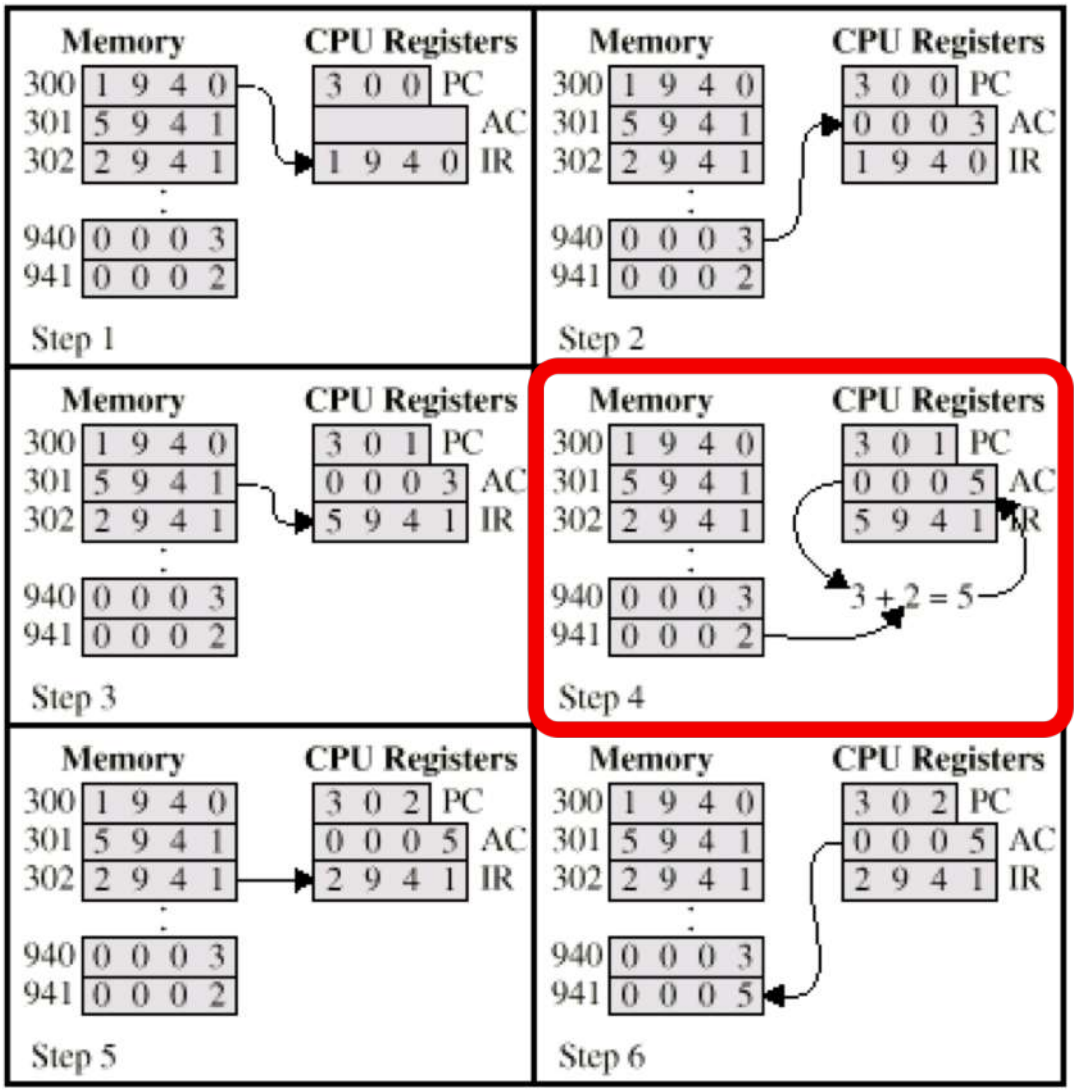
- Ejemplo completo
  3) El registro PC se incrementa en uno (ahora 301) y se capta la siguiente instrucción



1.3 – Arquitectura
Funcionamiento

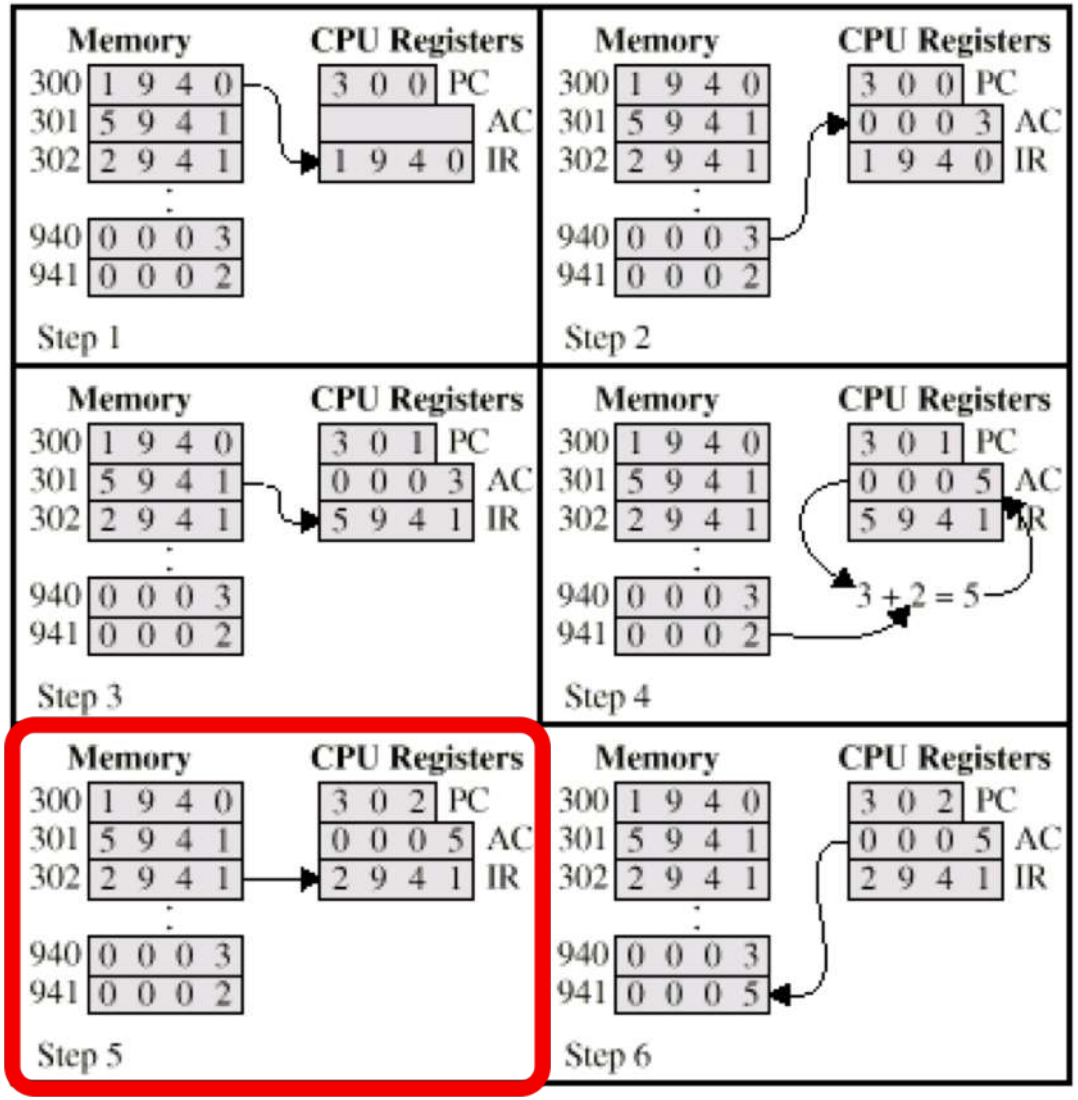
- Ejemplo completo

4) Los primeros 4 bits (el 5 decimal, → 0101 en binario) indican la instrucción.
Esta indica que hay que sumar el acumulador con el contenido de la memoria en la dirección 941 (es decir 1001 0100 0001 en binario). El contenido de AC y el de la dirección 941 se suman y el resultado se almacena en AC.



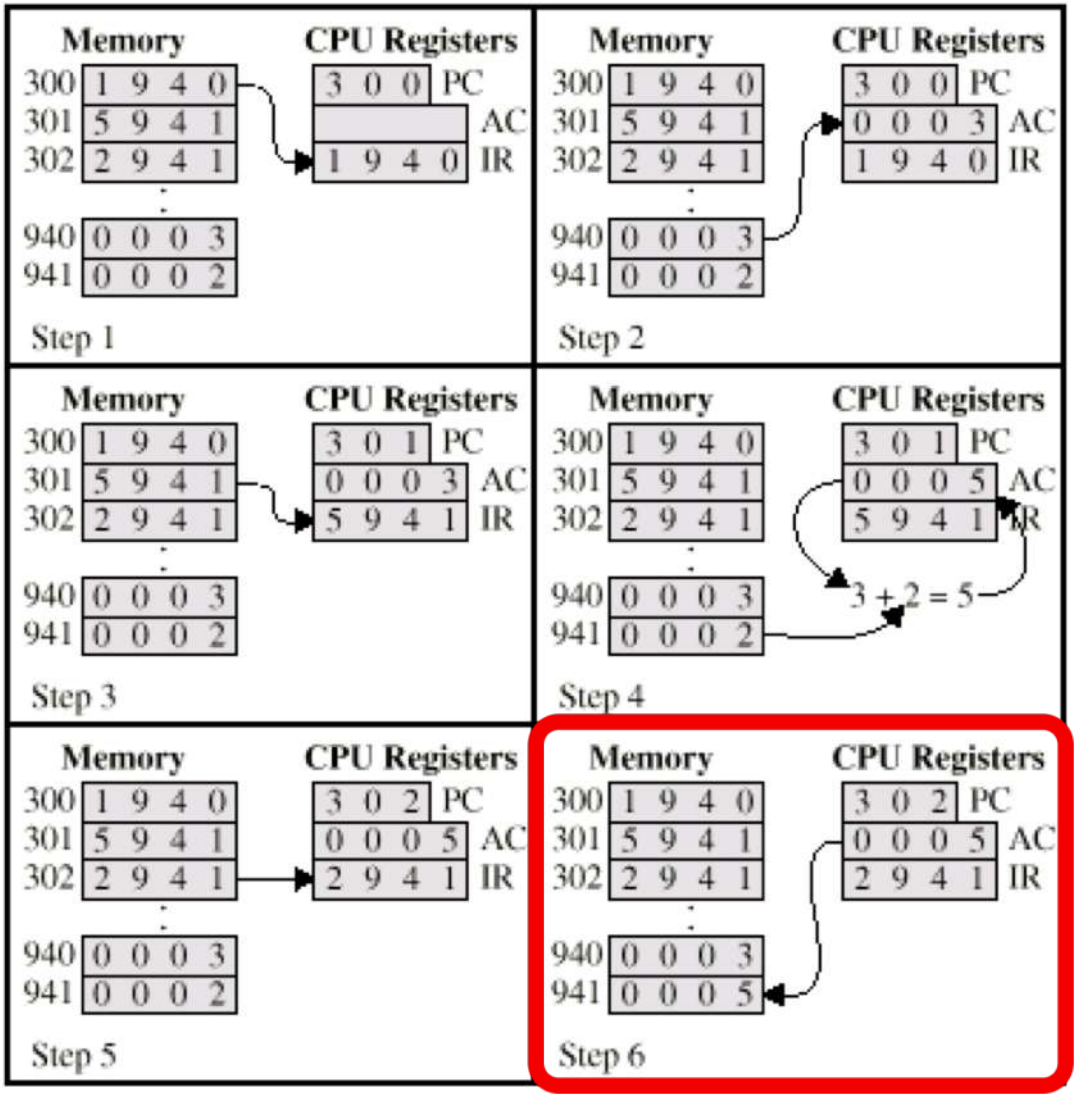
1.3 – Arquitectura
Funcionamiento

- Ejemplo completo
  5) El registro PC se incrementa en uno (ahora 302) y se capta la siguiente instrucción



1.3 – Arquitectura
Funcionamiento

- Ejemplo completo
  6) Los primeros 4 bits (el 2 decimal 0010 binario) indican que el contenido del acumulador se debe almacenar en la memoria en la dirección 941 (es decir 1001 0100 0001 en binario)..



Arquitectura i Configuracions Informàtiques

Tema 1. Introducción – Parte 2

Davide Careglio