

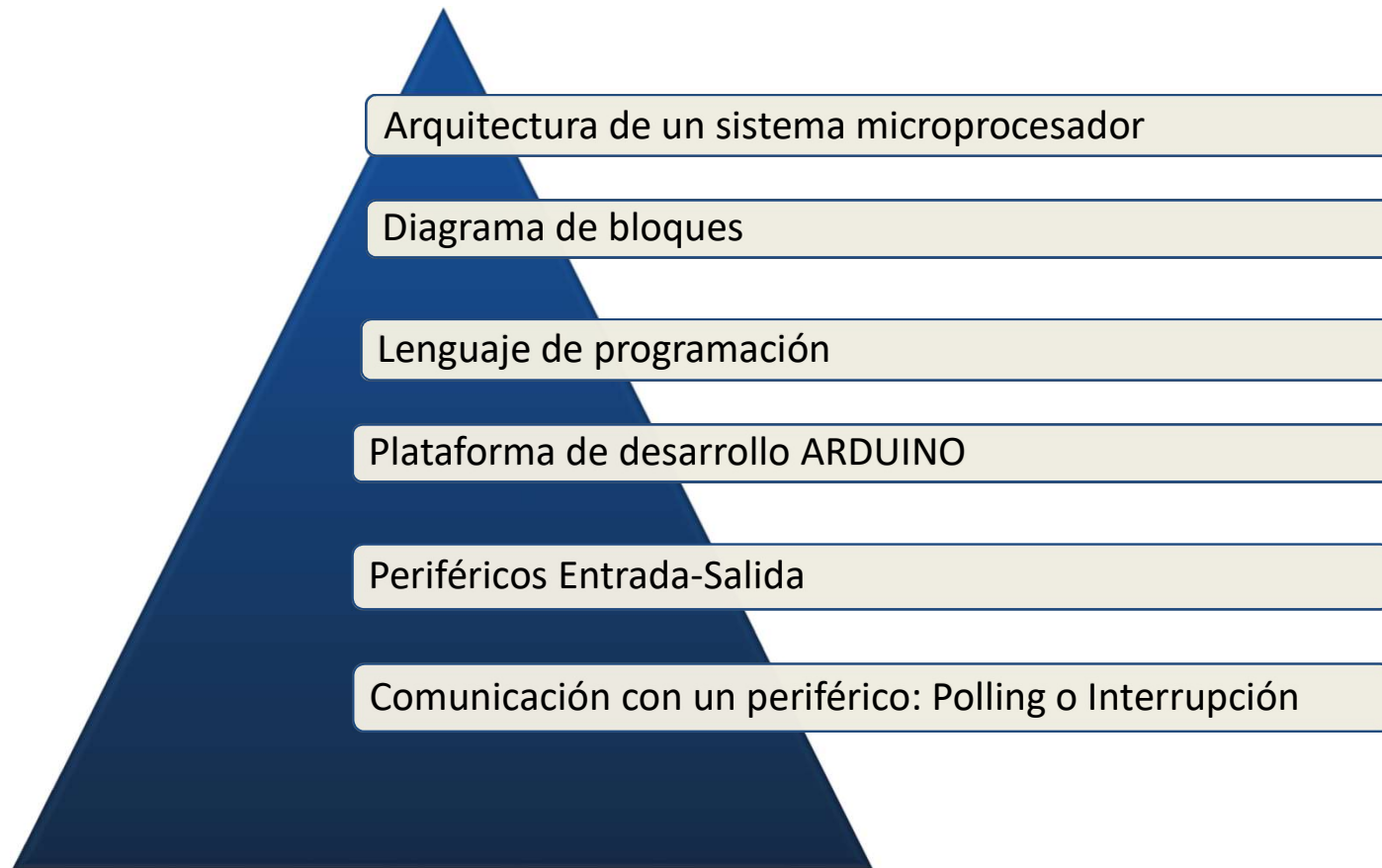


Tema 1-Sistemas Basados en Microprocesadores

Microprocesadores



Sistemas basados en microprocesadores





Procesadores Digitales

Teoría

Manuel Lamich Arocas
Despacho: TR2 209
@: manuel.lamich@upc.edu

Prácticas

Grupo 111	Grupo 112	Grupo 113
Manuel López Palma		Manuel Lamich Arocas
Despacho: TR2 240		Despacho: TR2 209
@: mlopezpalma@mlopezpalma.com		@: manuel.lamich@upc.edu

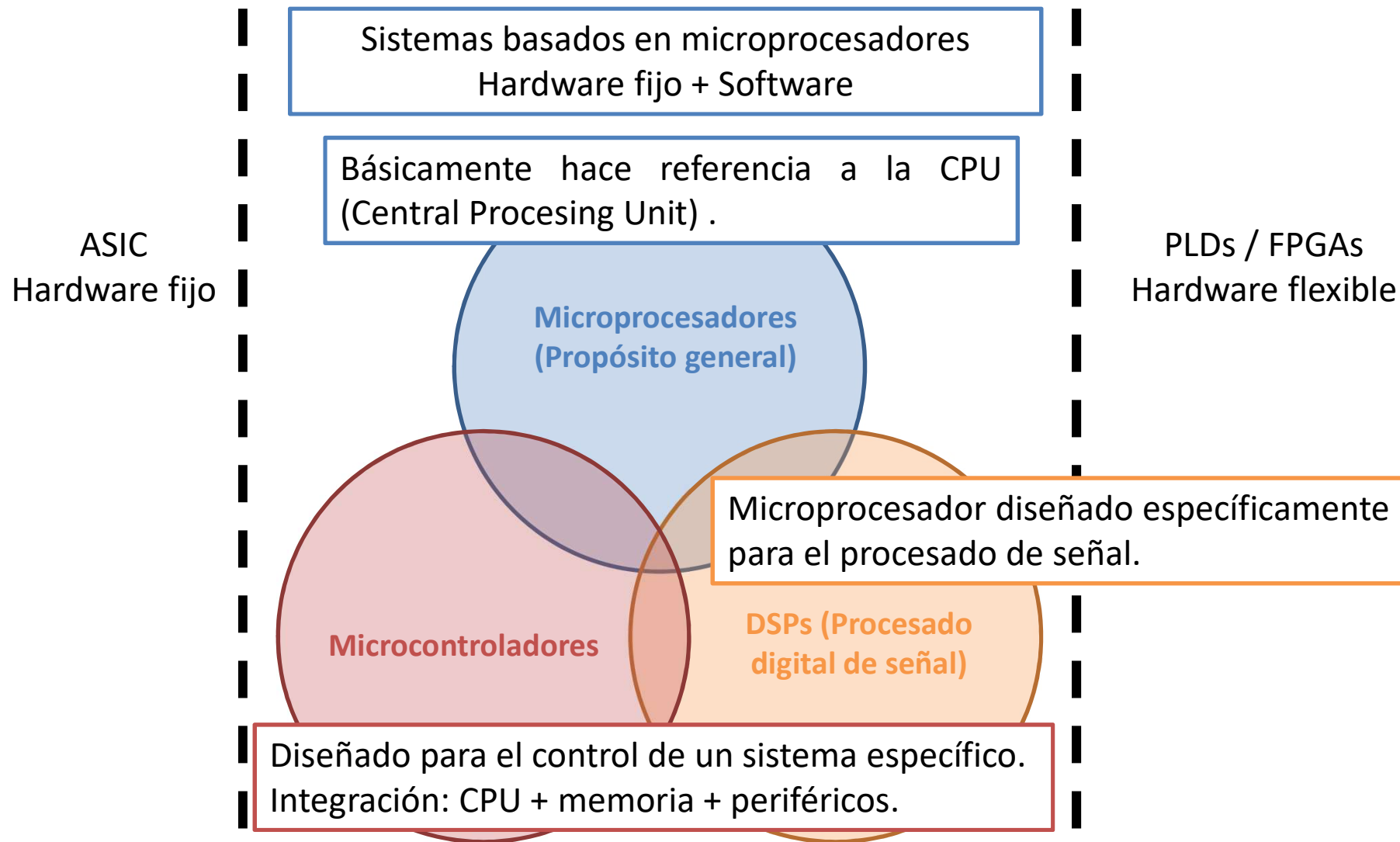


Evaluación

1er examen	-----	15%
2º examen	-----	25%
Laboratorio	-----	60%



Sistemas basados en microprocesadores





Elementos constituyentes de un sistema microprocesador

- Tres partes claramente diferenciadas:

Hardware

Circuitos físicos que implementan el sistema: CPU, Memoria, Periféricos, Buses.

- Estructura Von Neumann
- Estructura Harvard

Software

Conjunto de instrucciones que ejecuta el sistema.

- Programas

Firmware

Es un software grabado en la estructura electrónica del sistema microprogramable.

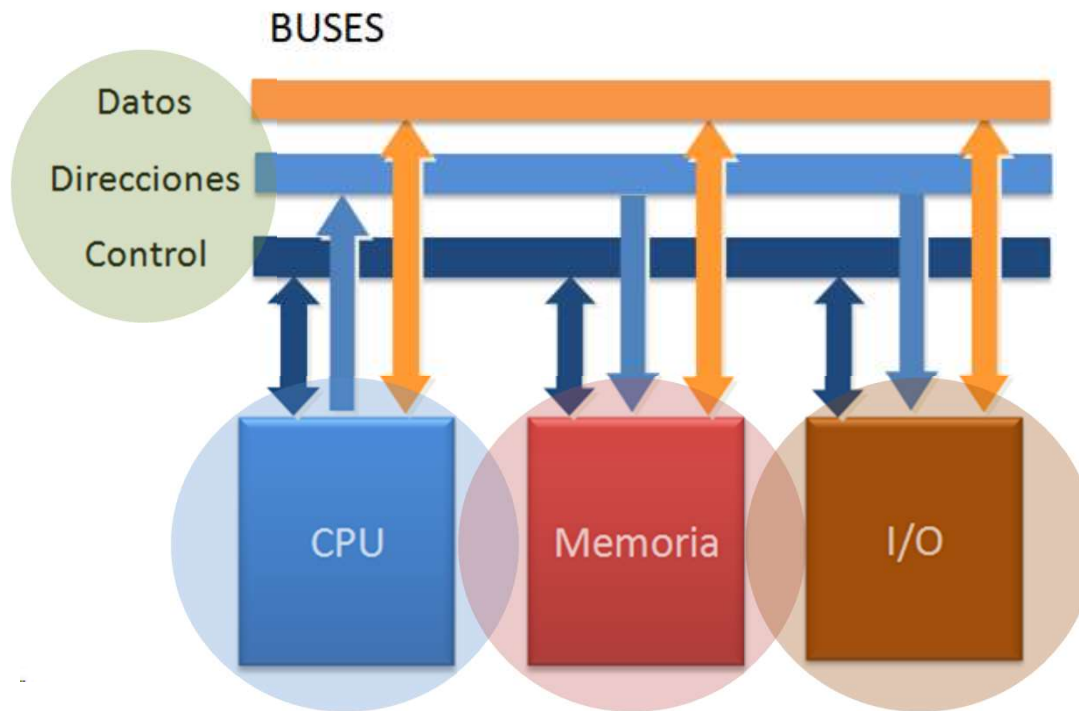
- BIOS (Basic Input/Output System) en los PCs



Arquitectura de un sistema microprocesador



Diagrama de bloques



CPU (Central Processing Unit): Es el núcleo del sistema y controla el resto de componentes.

Memoria: Almacena las instrucciones y los datos necesarios para la CPU.

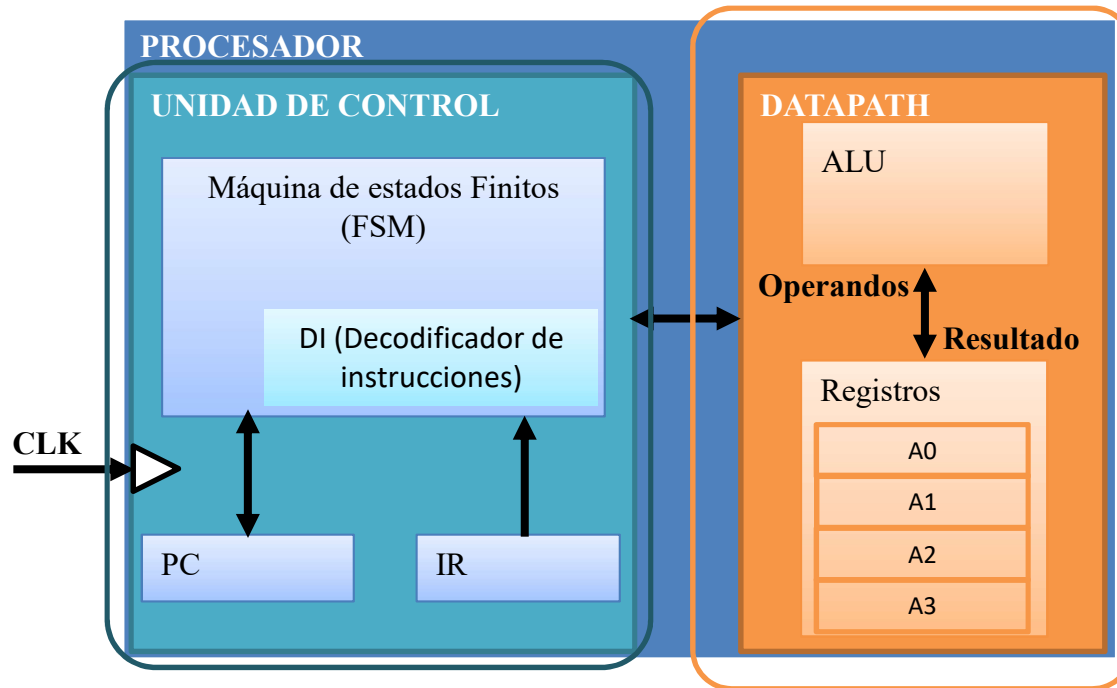
Periféricos entrada/salida: Permiten la comunicación del sistema con el exterior.

Buses del sistema (Bus de control, Bus de direcciones y Bus de datos): Permiten la conexión de la CPU con el resto de componentes

Arquitectura de un sistema microprocesador



CPU



ALU (Aritmetic Logic Unit): Esta formada por un circuito combinacional complejo que se encarga de realizar las operaciones aritméticas (suma, resta,...), lógicas (AND, OR,...), desplazamientos, etc.

Registros de propósito general: Son pequeñas memorias interna, donde se almacenan temporalmente los resultados intermedios de las operaciones.

CU (Control Unit): Se encarga de gobernar el funcionamiento global del sistema. Esta formada por un circuito secuencial complejo que se encarga de generar todas las señales de control necesarias, el contador de programa (PC) que indica en que posición de la memoria se encuentra la próxima instrucción a ejecutar, el registro de instrucciones (IR) que almacena la instrucción a ejecutar y el decodificador de instrucciones (DI), dispositivo que traduce las instrucciones del programa a microordenes para saber los pasos que se deben realizar para la ejecución de la instrucción.

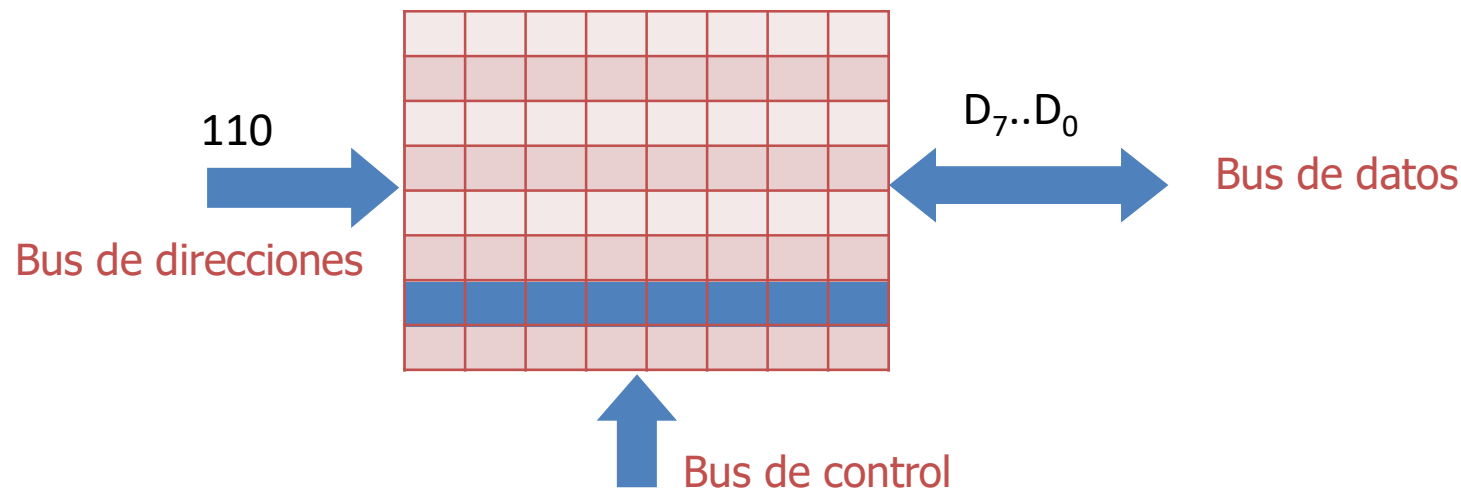


Arquitectura de un sistema microprocesador



Memoria

Circuito integrado que almacena información organizada en forma de arrays de bits.



Bus de direcciones: Selecciona el acceso a una determinada posición de memoria.
 2^n posiciones

Bus de datos: Dato a leer o a escribir. Tamaño de la palabra: m bits

Bus de control: Habilitación del integrado, selección de operación

Capacidad de la memoria: $2^3 \times 8 \text{ bits} = 64 \text{ bits} = 8 \text{ Bytes}$ ($2^n \times m \text{ bits}$)

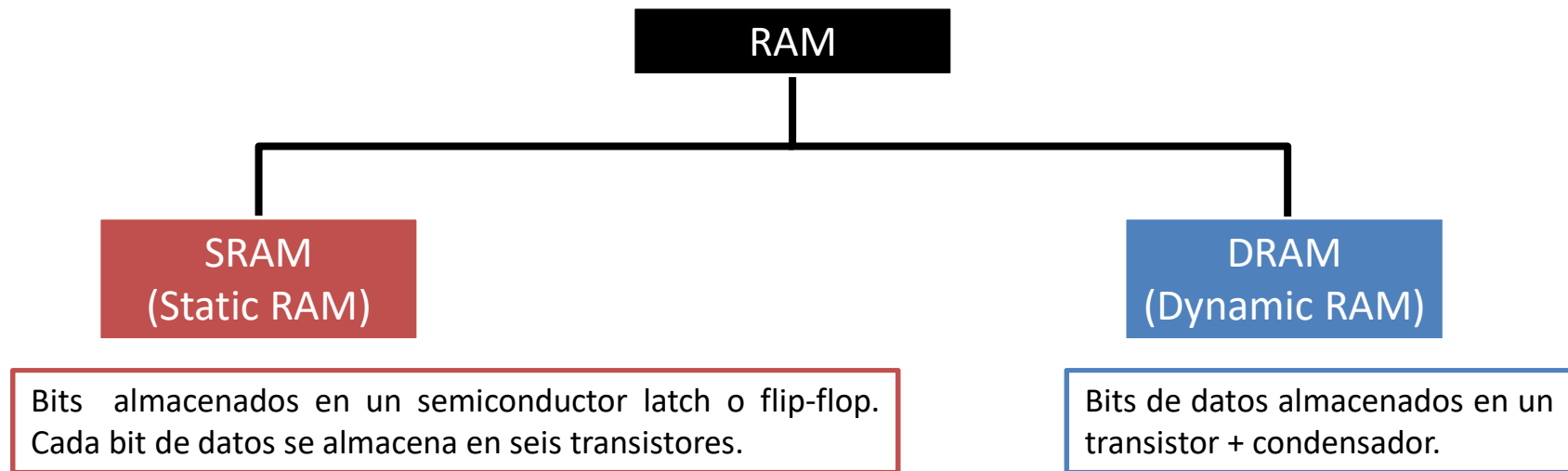


Arquitectura de un sistema microprocesador



Clasificación de las memorias en función de la volatilidad de los datos:

RAM (*Random Access Memory*): Volátiles



Memorias RAM estáticas (SRAM): No pierden información mientras están alimentadas, no es necesario refrescar la información almacenada. Memorias de datos en sistemas pequeños.

Memorias RAM dinámicas (DRAM): Necesidad de refresco para mantener la información. Mayor capacidad que SRAM y mayor tiempo de acceso que SRAM. Memorias de datos en sistemas grandes (PC's)

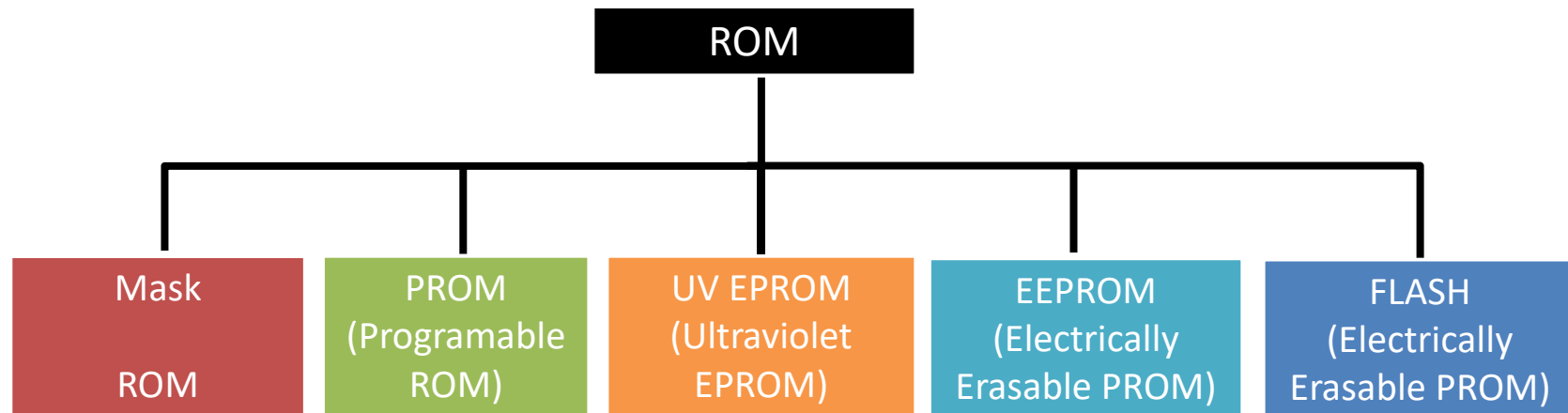


Estructura básica de un sistema microprocesador



Clasificación de las memorias en función de la volatilidad de los datos:

ROM (Read Only Memory): No volatiles



Memorias ROM

Los datos son permanentemente almacenados en la memoria durante el proceso de fabricación.

Memorias PROM (*Programmable ROM*)

Usuario define contenido mediante un programador. “Funde” fusibles
Solamente se pueden programar una vez (*OTP*)





Arquitectura de un sistema microprocesador



Clasificación de las memorias en función de la volatilidad de los datos:

ROM (Read Only Memory): No volatiles

Memorias UV EPROM (*Ultraviolet Erasable Programmable ROM*)

Programación igual que PROM

Posibilidad de restaurar fusibles. Borrado mediante luz Ultravioleta



Memorias EEPROM (Electrically Erasable and Programmable ROM)

Se programan y se borran de forma eléctrica

Se pueden programar y borrar desde la aplicación final

Útiles para guardar parámetros de configuración

Tiempo de lectura rápido. Como memoria RAM

Tiempo de escritura lento. Orden milisegundos

Memorias FLASH

Similares a las memorias EEPROM

Tiempo de escritura mucho menor que EEPROM

Mayor capacidad de almacenamiento que EEPROM





Puertos de entrada/salida

- Permiten la comunicación del sistema con el exterior.
 - Los microcontroladores llevan incorporados periféricos sencillos.
 - Periféricos de carácter digital y de carácter analógico.
-
- **Ejemplos de periféricos:**
 - Pulsadores, Finales de carrera, Teclados, Puertos para comunicaciones (RS-232, CAN, ETHERNET,...), Indicadores luminosos (Diodos Led), Display Numéricos, Visualizadores alfanuméricos, Temporizadores, Convertidores A/D y D/A,



Buses

- Bus de direcciones
 - Selecciona la dirección de memoria o I/O con la que se intercambia la información.

- Bus de datos
 - Líneas por las que se intercambia la información.

- Bus de Control
 - Señales auxiliares de gobierno y sincronización



Clasificación

Según su funcionalidad:

- **CISC (Complex Instruction Set Computer):** Poseen un juego complejo de cientos de instrucciones, algunas de las cuales son muy complejas y potentes y requieren muchos ciclos de reloj del sistema para ejecutarse.
- **RISC (Reduced Instruction Set Computer):** Con un juego reducido de instrucciones, disponiendo de pocas instrucciones que se ejecutan normalmente en un ciclo de reloj y que permiten optimizar el hardware del sistema.

Según su arquitectura interna:

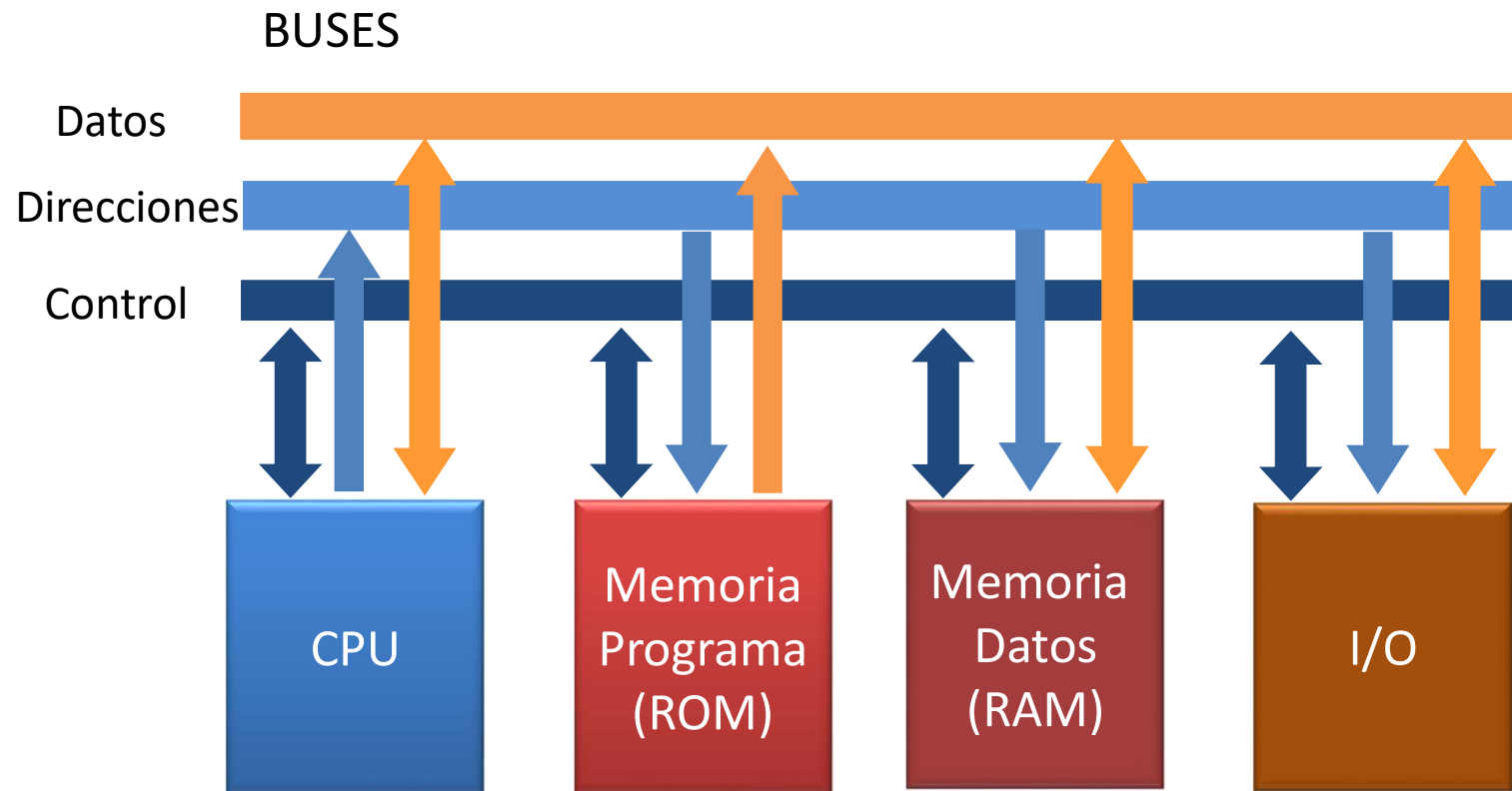
- **Von Neumann:** Se acceden a los datos y a las instrucciones a través del mismo bus de datos y direcciones.
- **Harvard:** Este modelo, tiene la CPU conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes.



Arquitectura de un sistema microprocesador



Estructura von Neumann



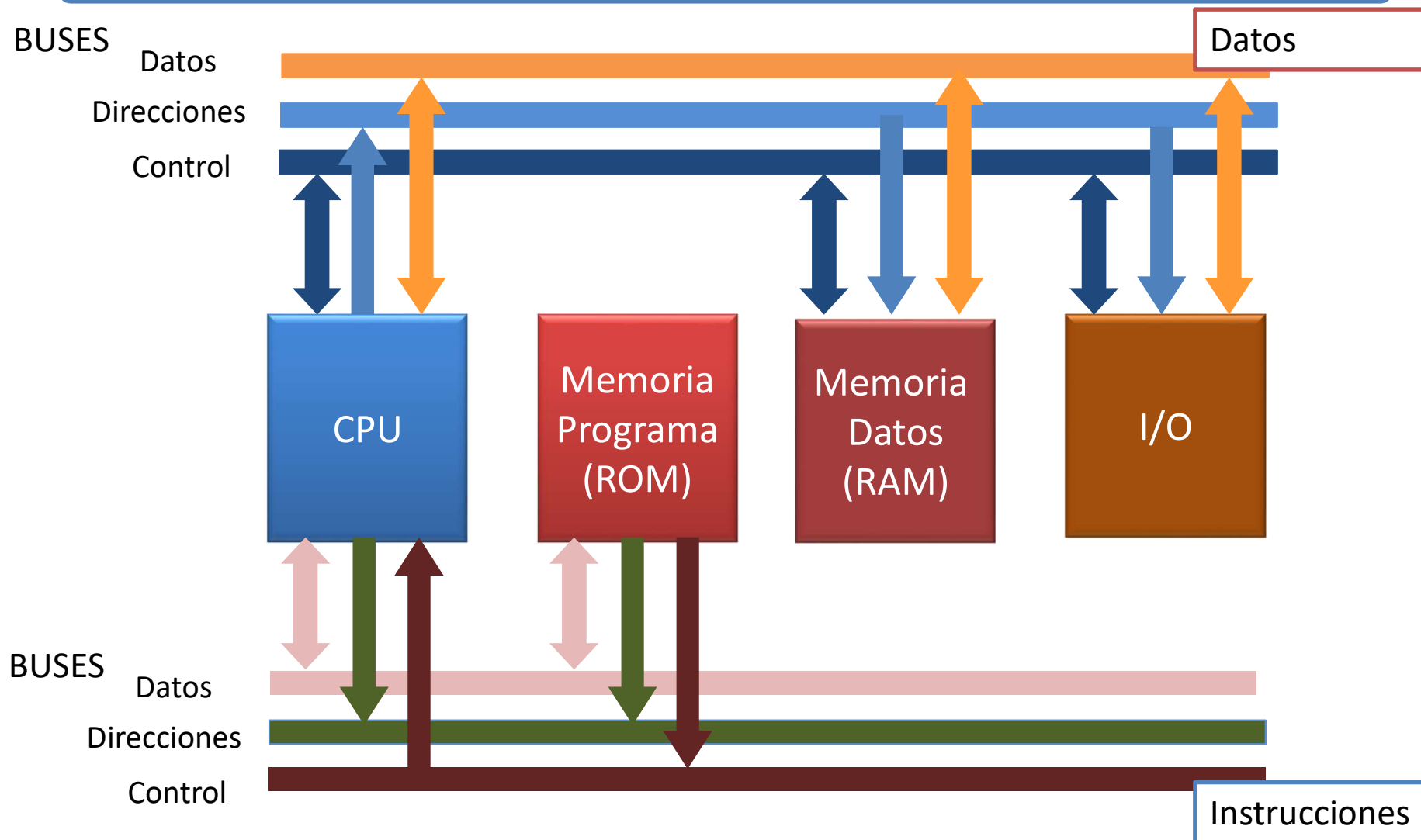
Las instrucciones y los datos son almacenados en memorias a las que accede la CPU a través de un único bus de direcciones, de datos y de control.



Arquitectura de un sistema microprocesador

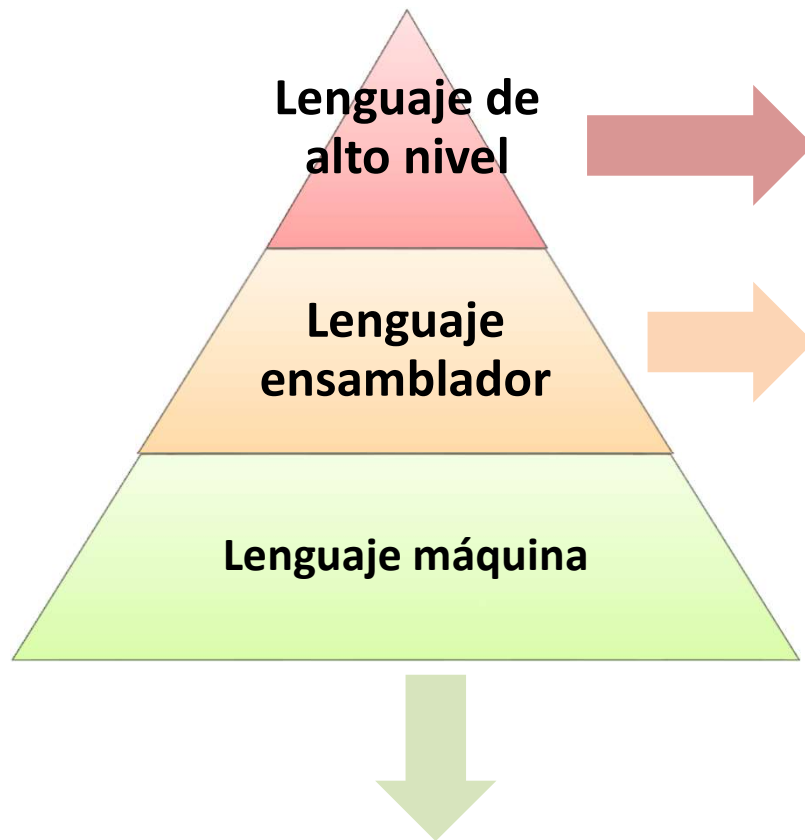


Estructura Harvard





Lenguajes de programación



La mayoría de la programación se hace en un lenguaje de alto nivel, que se puede ejecutar en diferentes máquinas. Es más fácil de escribir y mantener programas de alto nivel

El lenguaje ensamblador fue desarrollado para hacer una interfaz más sencilla entre la CPU y el programador, es una traducción directa del lenguaje máquina mediante literales.. Un programa en lenguaje ensamblador es más eficiente, aunque debe ser escrito para un procesador específico y requiere más tiempo para el desarrollo de programas.

El lenguaje máquina consiste en especificar las instrucciones a ejecutar mediante 0's y 1's, es el lenguaje que entiende la CPU.



Programa Compilador

Los lenguajes de alto nivel se pueden ejecutar en cualquier máquina, el código fuente se convierte en código máquina mediante un compilador.





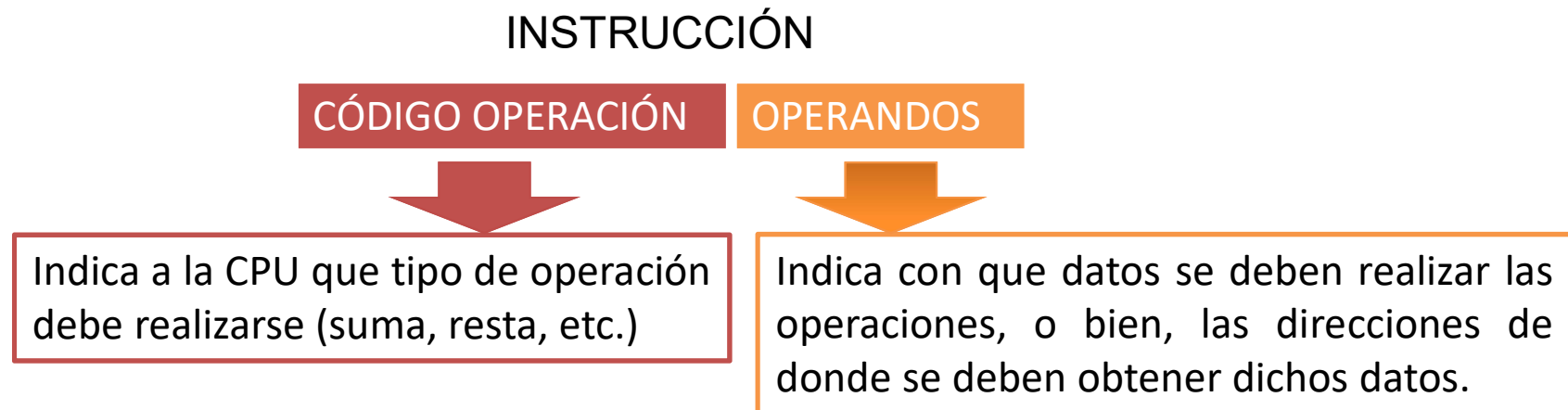
Programa Ensamblador

El lenguaje ensamblador debe ser escrito para un procesador específico, esto implica que el programador debe entender la estructura de registros del procesador. El programa ensamblador convierte el código fuente en código máquina.





Formato de una instrucción



Clasificación de las instrucciones:

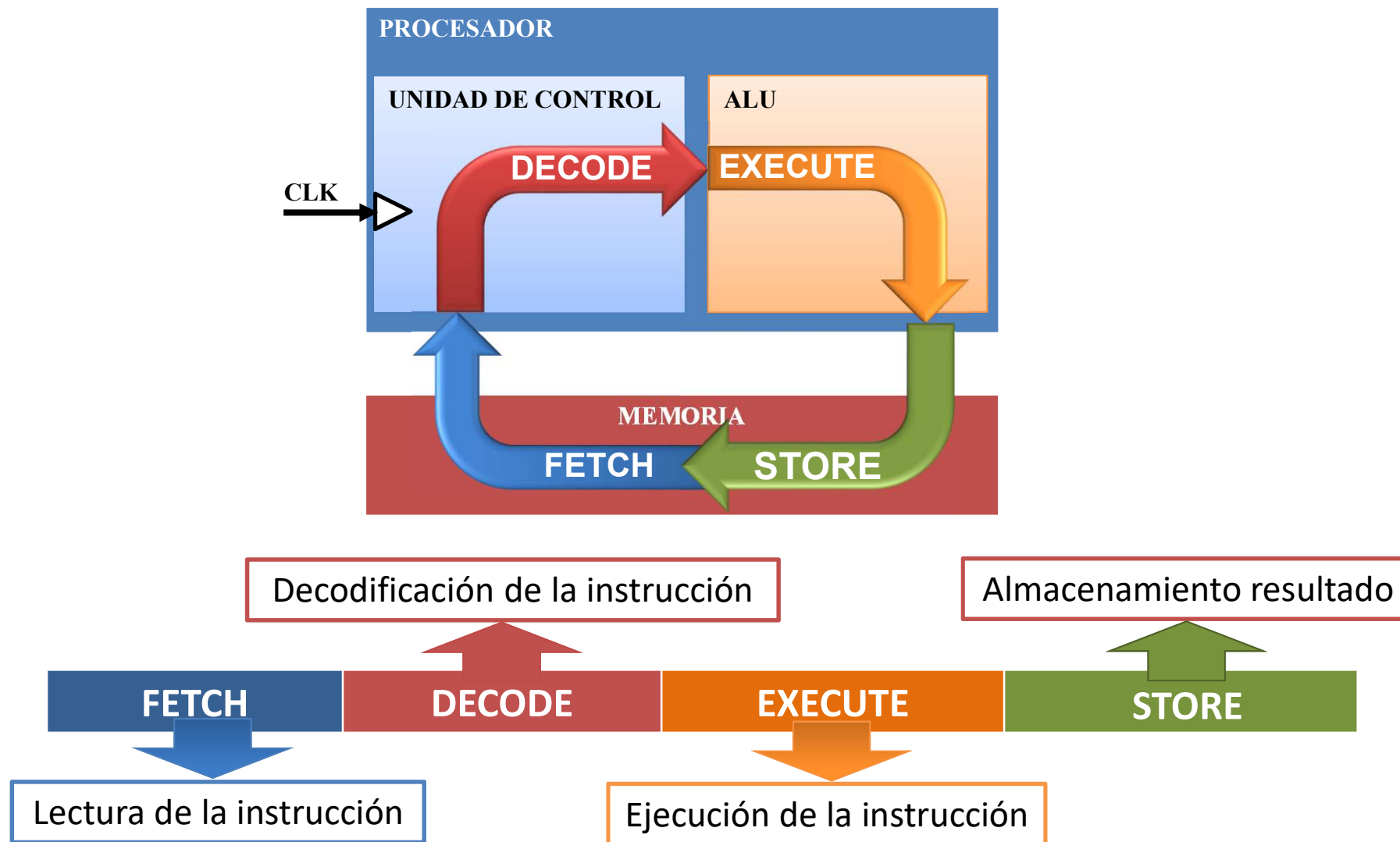
- **Instrucciones aritméticas/lógicas:** Realizan las operaciones aritméticas o lógicas entre dos operandos, tales como la suma, división, suma lógica, etc.
- **Instrucciones de transferencia de datos (entrada/ salida):** Estas instrucciones "ordenan" el paso de información entre distintos dispositivos del sistema.
- **Instrucciones de control:** Controlan el flujo del programa. Este tipo de instrucciones corresponde a los saltos condicionales o incondicionales .



Funcionamiento de un microprocesador

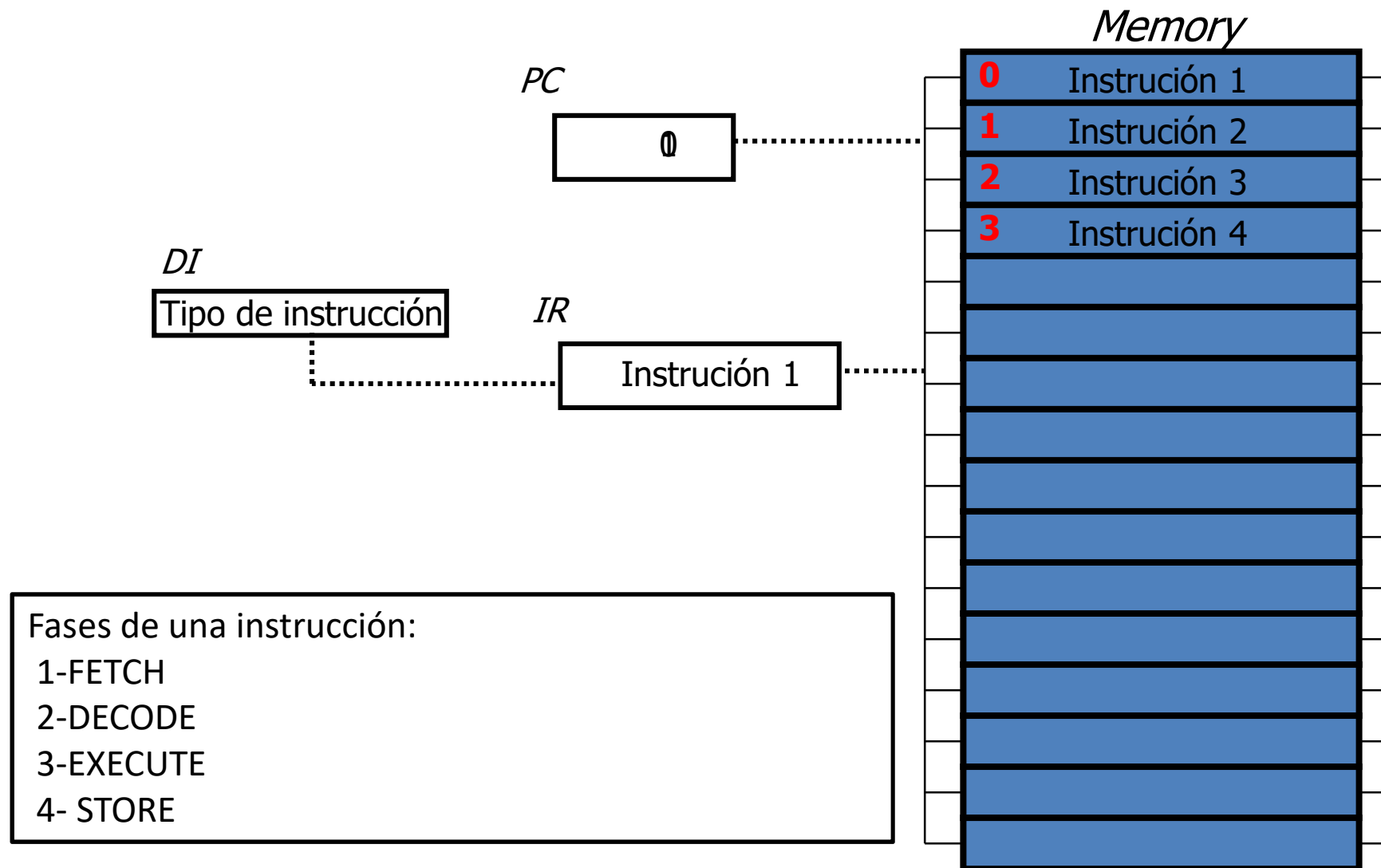


Fases de ejecución de una instrucción:





Funcionamiento de un microprocesador

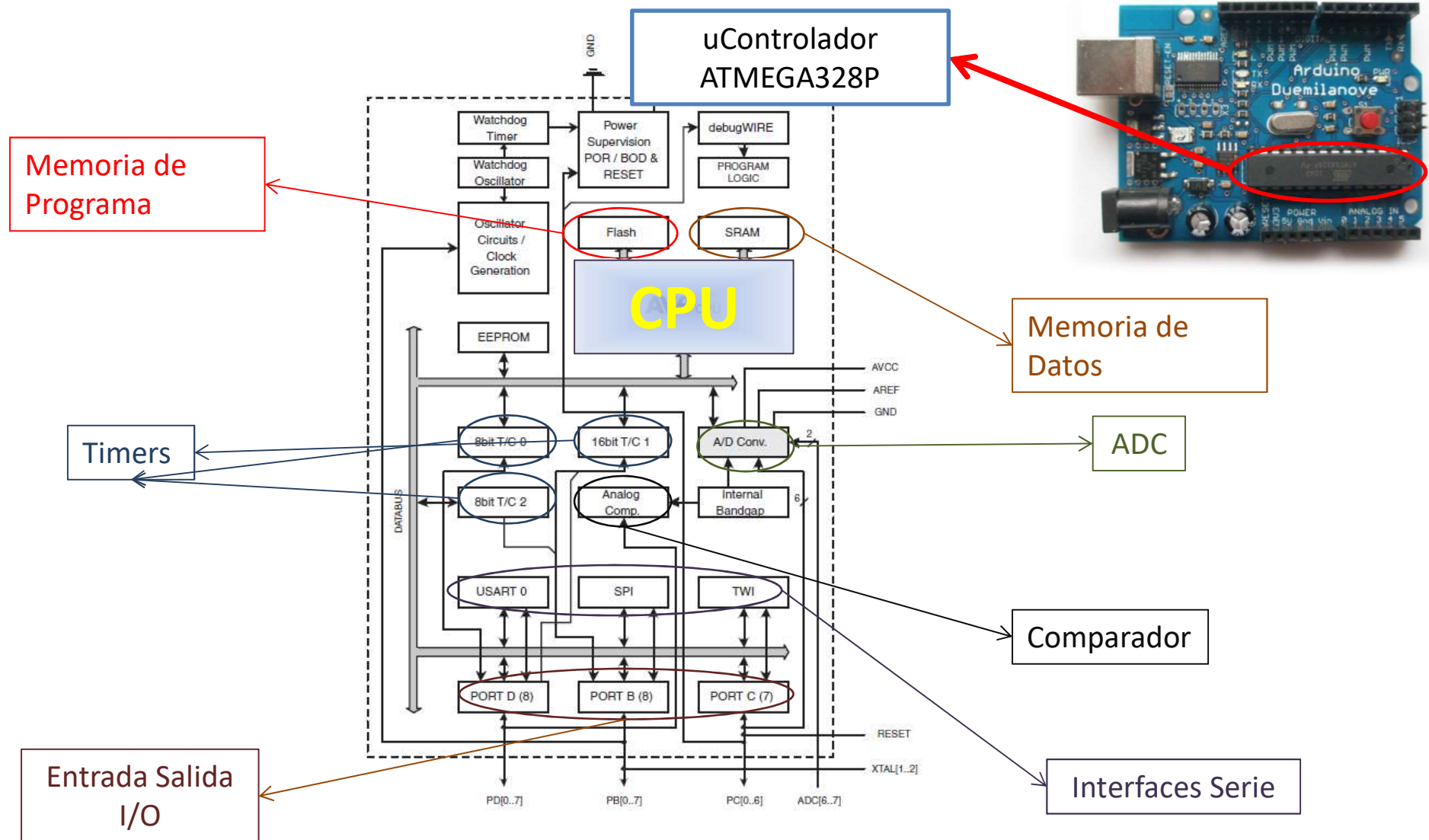




Plataforma de desarrollo ARDUINO



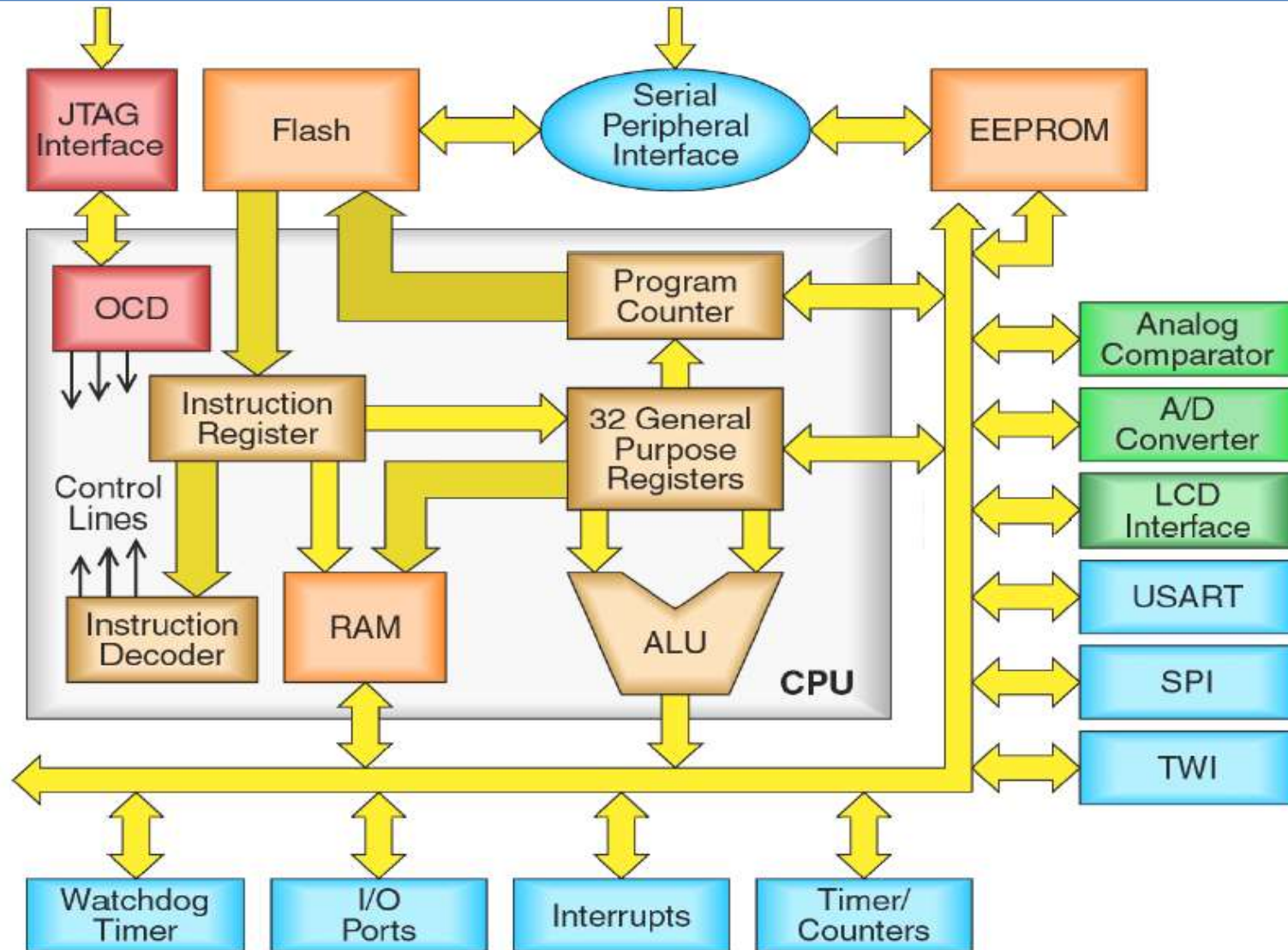
Diagrama de bloques del Microcontrolador ATMEGA328P



Plataforma de desarrollo ARDUINO



Diagrama de bloques del Microcontrolador ATMEGA328P





Comunicación con un periférico



- Periféricos de muy diversa naturaleza
 - Necesidad de homogeneizar la comunicación con los diversos periféricos.
- Controlador de periférico
 - Interfaz con periférico: Circuitos de activación y control.
 - Interfaz con CPU: Registros para la comunicación con periférico.



Sincronización entre CPU y periféricos

- La transferencia entre la CPU y los periféricos no se puede realizar en cualquier momento.
- CPU y periféricos trabajan de forma independiente (no sincronizada).
- No se pueden transmitir datos hasta que ambos están listos.
- Para sincronizarlos existen dos técnicas:
 - Sincronización mediante encuesta “Polling”
 - Sincronización mediante interrupción



Técnicas de sincronización

- Sincronización por encuesta o polling.
 - La CPU interroga al periférico para determinar si está listo para realizar la transferencia.
 - La CPU está a la espera de que el periférico esté listo.
- Sincronización por interrupción.
 - La CPU ejecuta su programa. Cuando el periférico está listo se lo indica, Interrumpe a la CPU. Entonces la CPU abandona su tarea y atiende al periférico.



Sincronización por encuesta o polling

- La CPU interroga al periférico para determinar si está listo para realizar la transferencia.
- La CPU está a la espera de que el periférico esté listo.

Ventajas:

- El control de flujo de la transferencia es totalmente por software (es mas sencillo programar el sistema).
- No es necesario Hardware adicional.

Inconvenientes:

- La CPU pierde mucho tiempo consultando a los periféricos (si sobra tiempo de proceso no es problemático).

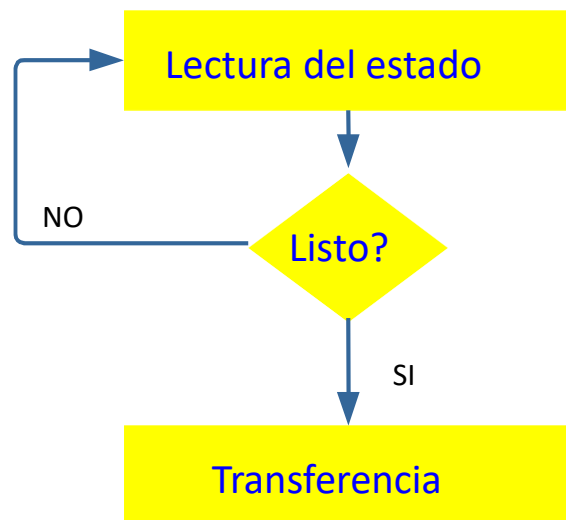


Periféricos Entrada-Salida (I/O)

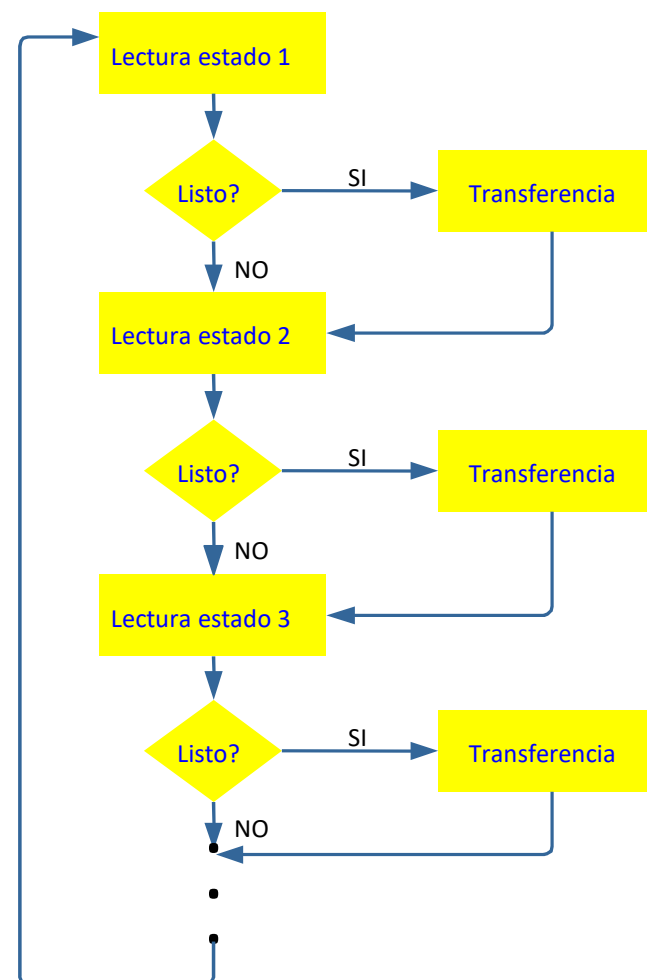


Sincronización por encuesta o polling

Algoritmo con un periférico

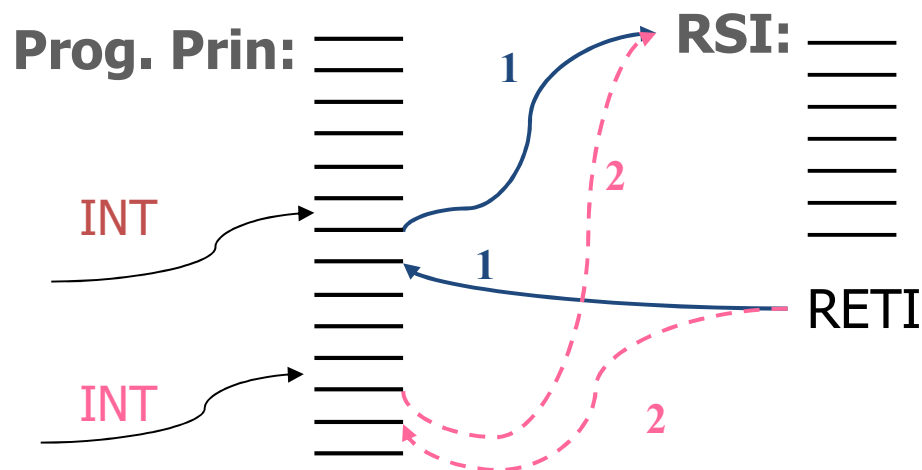


Algoritmo con varios periféricos





Sincronización por interrupción



1. La CPU ejecuta su programa. Cuando el periférico está listo se lo indica, Interrumpe a la CPU. Entonces la CPU abandona su tarea y atiende al periférico.
2. La CPU pasa a ejecutar la rutina de Servicio de Interrupción (RSI).
3. Una vez finalizada esta rutina retorna a la ejecución del programa principal en el punto donde lo había dejado.



Sincronización por interrupción

- Para conocer el punto donde debe regresar la CPU guarda automáticamente esta dirección antes de ejecutar la RSI.
- Las RSI finalizan siempre con la instrucción de retorno de interrupción (RETI).

Ventajas:

- La CPU puede realizar otras tareas mientras el periférico no está listo.

Inconvenientes:

- La programación de la CPU es más compleja ya que no se conoce a priori, el flujo de ejecución de instrucciones.
- Es necesario utilizar Hardware adicional.