

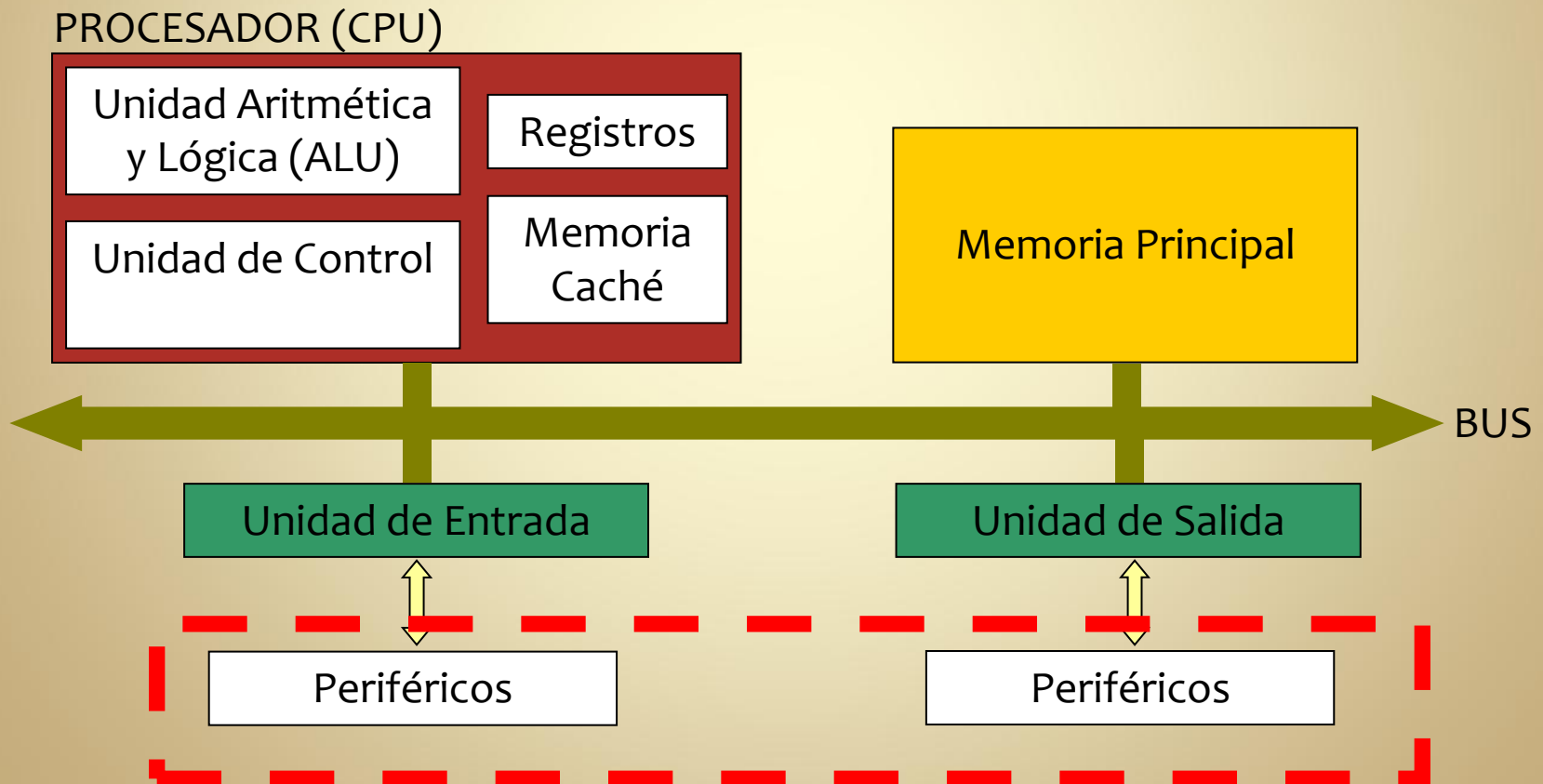
ORGANIZACIÓN DEL COMPUTADOR (MODELO DE VON NEUMANN)

Principios de Computadores

1. **Arquitectura Von Neumann**
2. **Funcionamiento del ordenador de Von Neumann**
3. **Ejemplo: movimiento de instrucciones**
4. **Ejemplo: ejecución de una instrucción**
5. **Proceso Von Neumann**
6. **Proceso Harvard**
7. **Conclusiones**

1. Arquitectura Von Neumann

Los conceptos de esta arquitectura fueron publicados en 1945 por John Von Neumann. Esta arquitectura es la base de la mayoría de las arquitecturas actuales.



PERIFÉRICOS

Conjunto de componentes usados por el computador para diversos fines:

- Interacción con el usuario (teclado, ratón)
- Comunicación externa (módems, tarjetas de red)
- Almacenamiento de información (discos duros, unidades magnético-ópticas)

PROCESADOR (CPU)

Unidad Aritmética
y Lógica (ALU)

Registros

Unidad de Control

Memoria
Caché

Memoria Principal

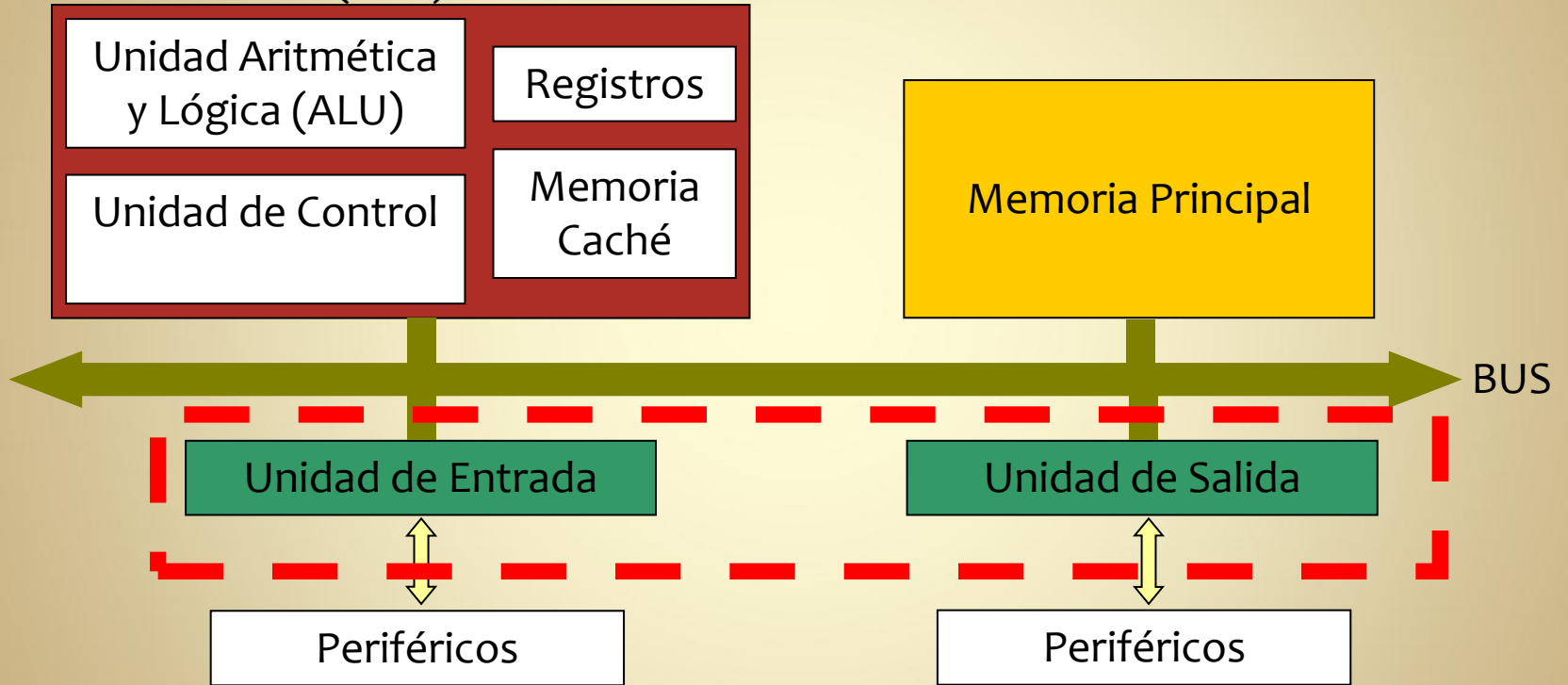
BUS

Unidad de Entrada

Unidad de Salida

Periféricos

Periféricos



UNIDAD DE ENTRADA

- *Unidad que gestiona la transferencia de los datos y las señales de control entre los periféricos de entrada y el procesador.*
- El computador acepta la información codificada a través de las unidades de entrada (ej: teclado: al pulsar una tecla se traduce ese valor alfanumérico automáticamente a su código binario).
- Más dispositivos de entrada: ratones, mandos de juegos, micrófonos, etc

UNIDAD DE SALIDA

- *Unidad que gestiona la transferencia de los datos y las señales de control entre el procesador y los periféricos de salida.*
- Dispositivos de salida: impresora, pantalla, etc.

La unidad de E/S libera al procesador de gestionar cada periférico de forma independiente

PROCESADOR (CPU)

Unidad Aritmética
y Lógica (ALU)

Registros

Unidad de Control

Memoria
Caché

Memoria Principal

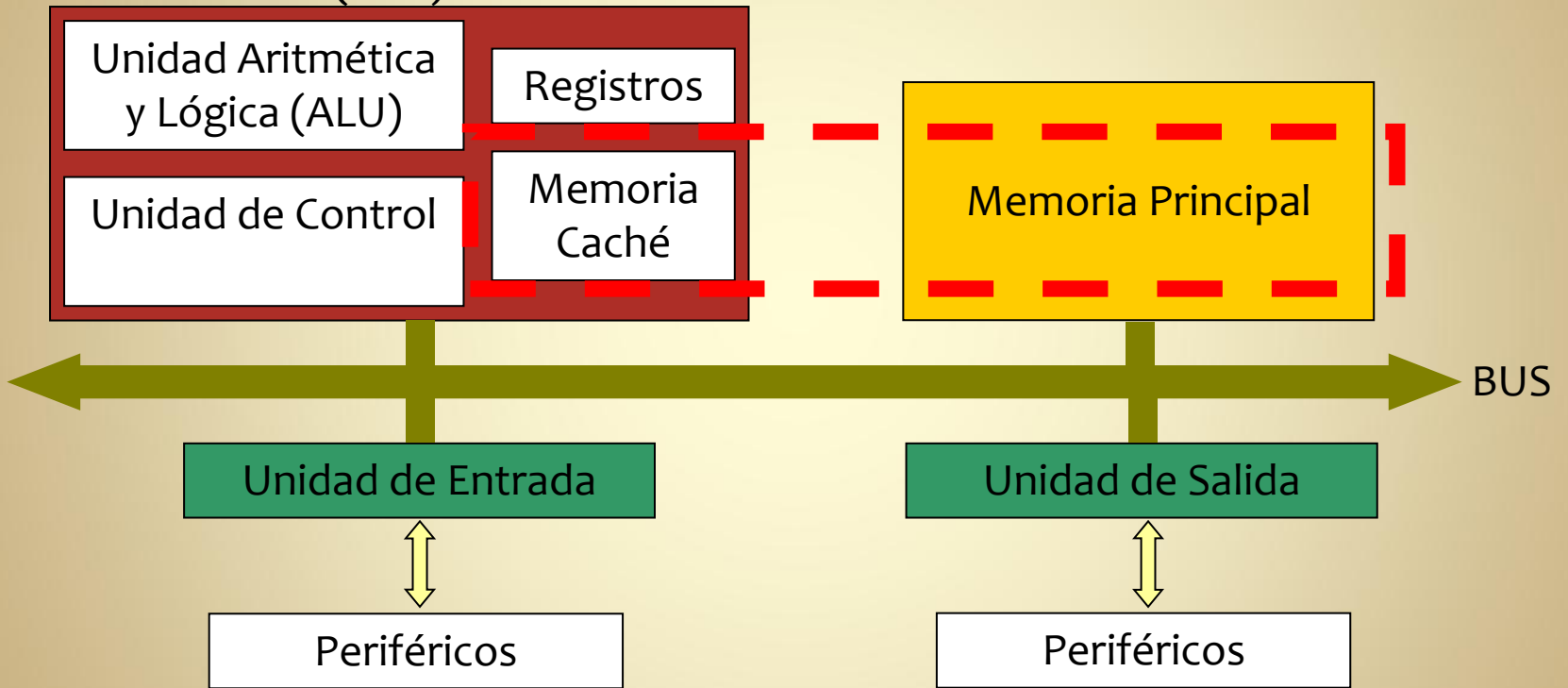
Unidad de Entrada

Unidad de Salida

Periféricos

Periféricos

BUS



MEMORIA

- Almacena: 1) los **datos leídos** y procesados por el computador, 2) las **instrucciones** (programas).
- Dos tipos de almacenamiento: primario y secundario.

Las instrucciones y los datos **comparten el mismo espacio de almacenamiento, siendo gestionados del mismo modo** por el sistema de memoria ->
CARACTERÍSTICA MUY IMPORTANTE DE LA
ARQUITECTURA VON NEUMANN

MEMORIA – Almacenamiento Primario

- Memoria rápida, que funciona a “velocidades electrónicas”.
- La memoria contiene un gran número de “celdas” de almacenamiento, cada una contiene un bit de información.
- Las celdas no se suelen escribir o leer individualmente. Se procesan en grupos de tamaño fijo: **PALABRAS** (n bits). El número de bits n de una palabra es la **LONGITUD DE PALABRA** (16 a 64 bits).
- La memoria se organiza de modo que el contenido de una palabra se puede almacenar o recuperar en una operación básica. A cada palabra se le asocia una **DIRECCIÓN**.

MEMORIA – Almacenamiento Primario

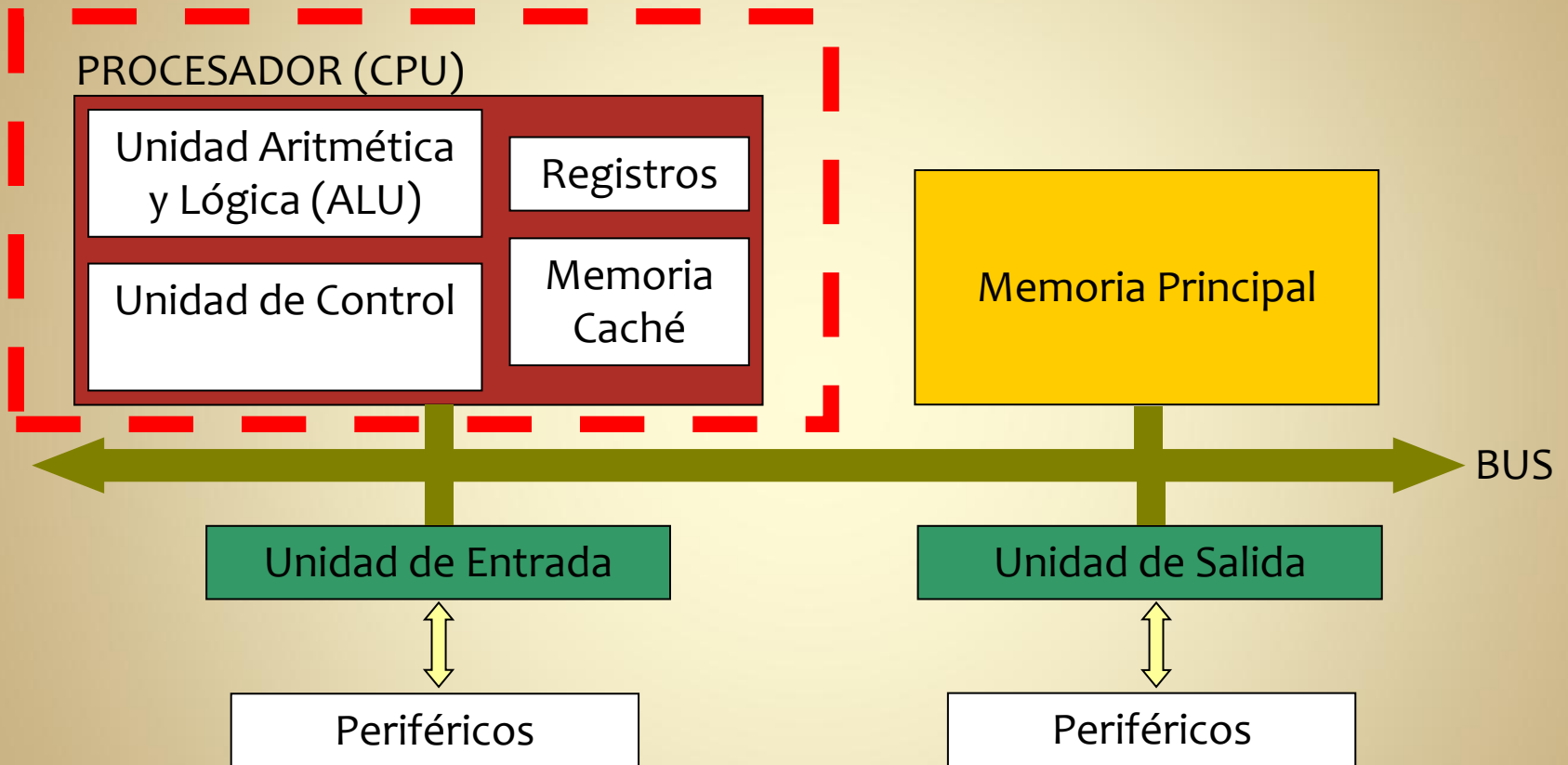
- Los programas a ejecutar deben residir en memoria durante su ejecución. Las instrucciones y los datos se escriben o leen en la misma memoria bajo el control del procesador.
- Objetivo: poder acceder a cualquier posición de memoria lo más rápido posible. *El tiempo requerido para acceder a la palabra* es el **tiempo de acceso a memoria**.
- **Memoria RAM** (memoria de acceso aleatorio): memoria en la que cualquier posición puede alcanzarse en un tiempo corto y fijo (pocos ns – 100 ns).

MEMORIA – Almacenamiento Primario

- La memoria de un computador normalmente tiene una **jerarquía de niveles** (3-4 niveles) de unidades RAM de distintos tamaños y velocidades.
- Las más pequeñas y rápidas son los **REGISTROS**. Fuertemente conectadas al procesador (estando incluso en el mismo chip para mejorar el rendimiento). La unidad más grande y lenta es la **MEMORIA PRINCIPAL**.
- El almacenamiento primario es esencial, pero costoso. Por eso es necesario el almacenamiento secundario.

MEMORIA – Almacenamiento Secundario

- Más barato que el primario cuando es necesario almacenar grandes cantidades de datos y muchos programas, o información a que se accede con poca frecuencia.
- Ejemplos: discos magnéticos, CD-ROM, etc

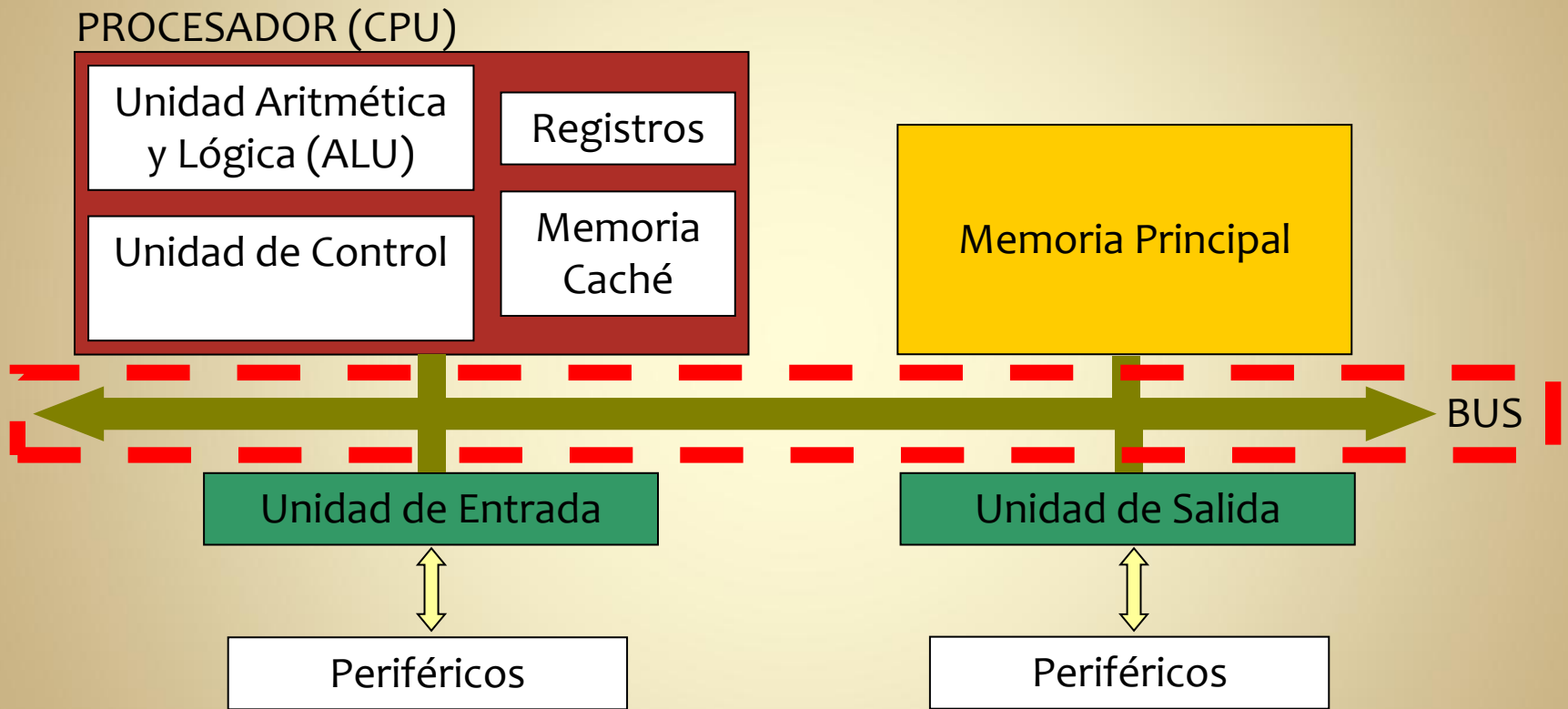


PROCESADOR (Unidad Central de Procesos-CPU)

- **Registros internos:** memoria muy pequeña (almacena una palabra) y muy rápida para el almacenamiento temporal de datos e instrucciones. Permiten reducir el número de accesos a memoria.
- **Memoria caché:** subconjunto del espacio de memoria principal con tiempo de acceso menor que el de la memoria principal.
- **Unidad Aritmético-Lógica (ALU):** encargada de realizar las operaciones aritméticas o lógicas sobre los datos que requieran las instrucciones.
- **Unidad de Control:** encargada de 1) coordinar todos los elementos anteriores para ejecutar las instrucciones, 2) generar las SEÑALES DE TEMPORIZACIÓN: señales que regulan las transferencias de información entre las distintas unidades.

PROCESADOR (Unidad Central de Procesos-CPU)

- Cada registro puede almacenar una palabra.
- El tiempo de acceso a los registros es más rápido que el tiempo de acceso a la unidad de memoria caché más rápida de la jerarquía de memoria del computador.
- Cualquier operación aritmética o lógica se inicia llevando los operandos requeridos al procesador, concretamente a la ALU, donde se realiza la operación.
- El resultado de la operación se puede almacenar en memoria o se retiene en el procesador para su uso inmediato (en los registros).
- La Unidad de Control genera las señales que indican cuándo debe hacerse una acción determinada. La circuitería de esta unidad está físicamente distribuida por toda la máquina. Un gran conjunto de líneas de control (cables) lleva las señales usadas para la temporización y la sincronización de eventos a todas las unidades.



BUS

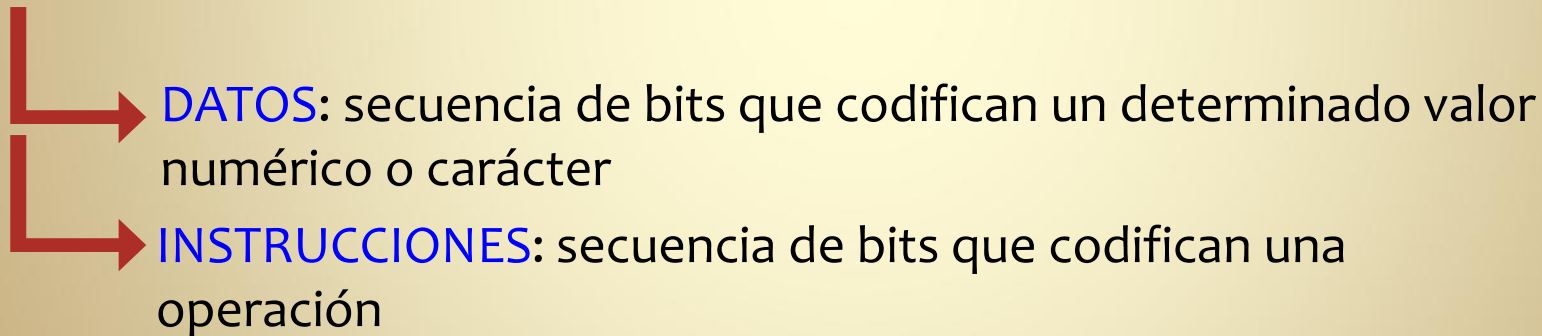
- En un computador todas las unidades pueden gestionar una palabra completa de datos a la vez.
- La palabra se transfiere enviando todos sus bits en paralelo a través de líneas (muchas líneas, 1 bit por línea). El grupo de líneas que conecta las distintas unidades es el BUS.
- El bus se compone de:
 - **Líneas datos** (bits de la palabra):
 - **Líneas de control**: Todas las unidades se conectan al bus, pero sólo dos pueden usar el bus a la vez en un momento dado para comunicarse. . Las líneas de control del bus se usan para recibir peticiones y conceder el uso del bus (**arbitraje**), indicar que se ha colocado un dato en el bus o se ha leído del bus, etc...
 - **Líneas de dirección**: indican la dirección de cada dispositivo conectado al bus.

BUS

- Los dispositivos conectados al bus no son todos igual de rápidos. Ej: impresora y teclado son lentos, CDs muy rápidos, procesador y memoria los más rápidos.
- Es necesario un mecanismo de transferencia eficiente, donde los dispositivos más rápidos no tengan que esperar por los más lentos: existen **registros buffer** en los distintos dispositivos que almacenan los datos (ej: buffer de la impresora) mientras el bus y el procesador pueden continuar con otras tareas.
- Existen sistemas con buses múltiples: mayor concurrencia de operaciones (dos o más transferencias en paralelo), pero aumenta la complejidad y el coste de la máquina.

2. Funcionamiento del ordenador de Von Neumann

- Principio de funcionamiento: **PROGRAMA** (*secuencia de INSTRUCCIONES que indican las acciones que debe realizar el procesador*).
- INSTRUCCIONES y DATOS se almacenan en una **MEMORIA COMÚN**.



- El modo de funcionamiento del computador consiste en la ejecución continua de instrucciones.

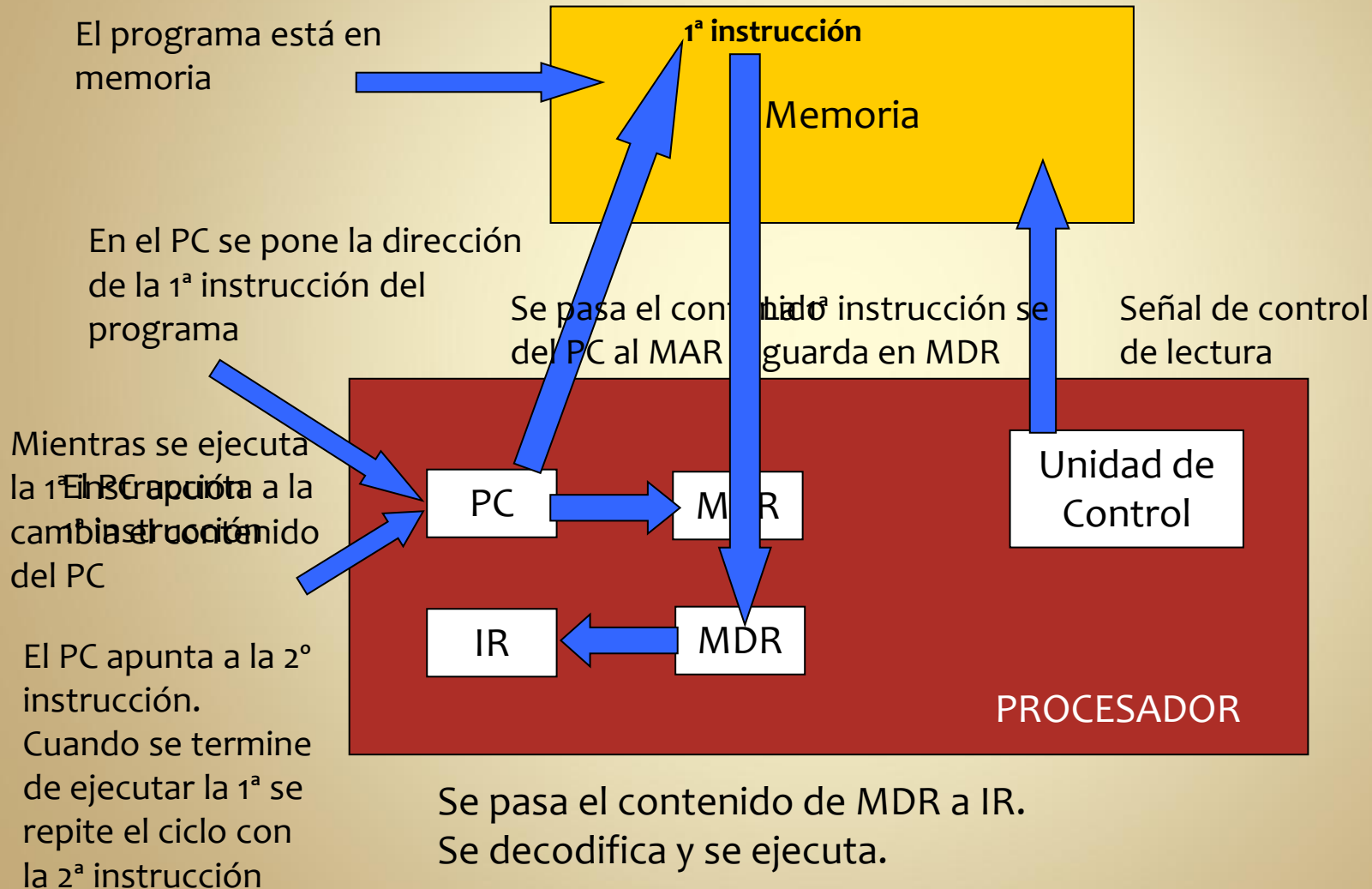
FASES DEL PROCESO DE EJECUCIÓN DE CADA INSTRUCCIÓN:

1. Lectura o carga de la instrucción: la instrucción se transfiere desde la memoria hasta un registro del procesador (***registro de instrucción***).
2. Decodificación de la instrucción: la unidad de control lee el contenido del registro de instrucciones y decodifica la operación asociada a la instrucción.
3. Ejecución de la instrucción: la unidad de control activa las señales necesarias para: 1) leer los operandos asociados a la acción (ej: de memoria a registros internos), 2) la ALU realiza la operación (ej: sumar dos registros), 3) almacenar el resultado (ej: en un registro, en memoria).
4. Determinación de la siguiente instrucción a ejecutar: la unidad de control determina tras la ejecución de una instrucción cuál es la posición de memoria que contiene la siguiente instrucción a ejecutar. Normalmente se ejecutan de forma ordenada, una detrás de otra (no siempre, ej. Instrucciones de salto).

REGISTROS DEL PROCESADOR:

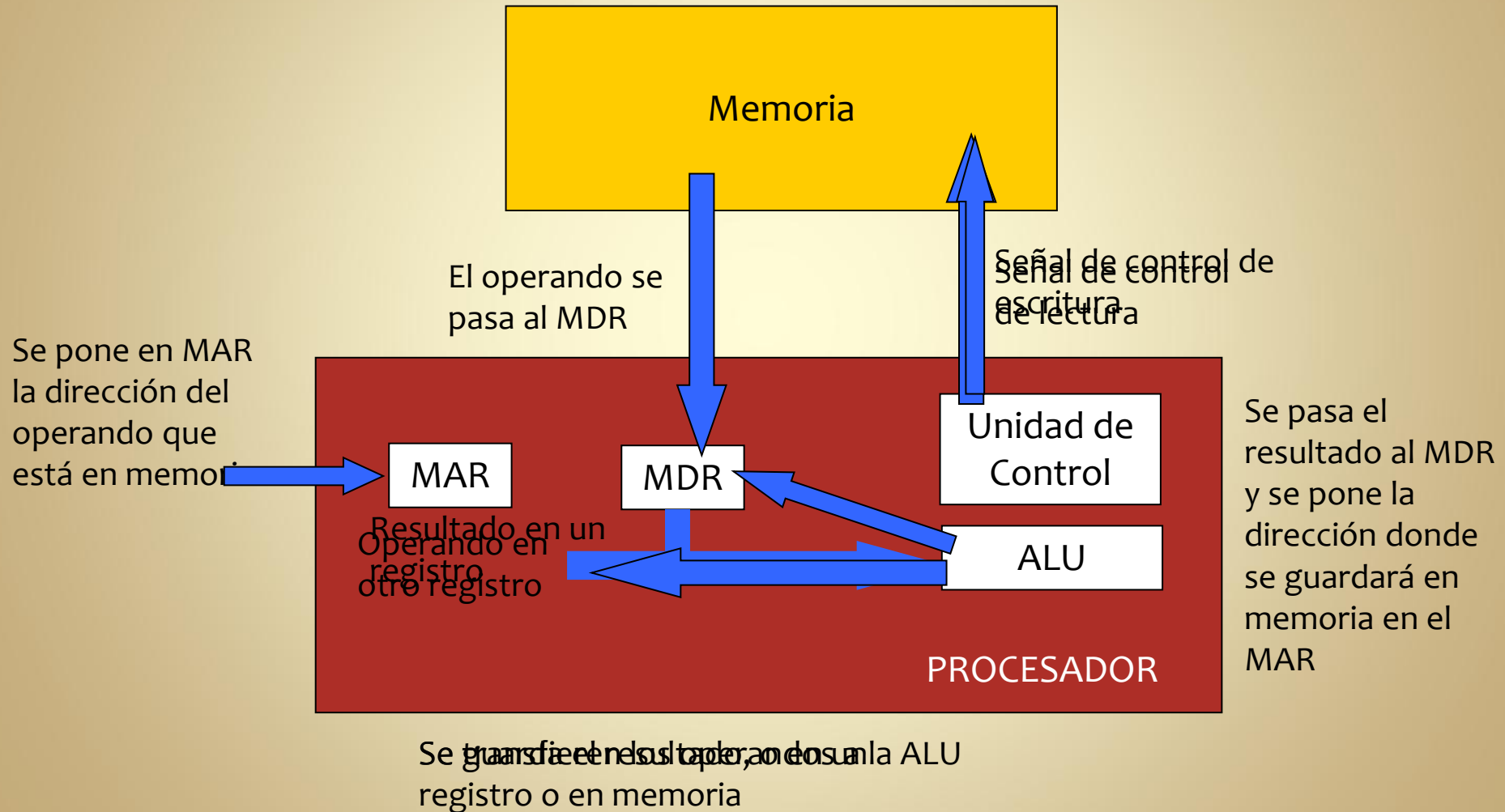
1. **Registro de instrucción (IR = *Instruction Register*)**: contiene la instrucción que se está ejecutando en ese momento.
2. **Contador de programa (PC = *Program Counter*)**: contiene la dirección de memoria donde está la siguiente instrucción que se ejecutará (PC “apunta” a la siguiente instrucción).
3. **Registro de datos de memoria (MDR = *Memory Data Register*)**: contiene el dato que se ha leído o que se va a escribir en una posición de memoria determinada.
4. **Registro de dirección de memoria (MAR = *Memory Address Register*)**: contiene esa dirección de memoria.
5. **Registros de uso general**: n registros ($R_0 \dots R_{n-1}$).

3. Ejemplo: movimiento de instrucciones



4. Ejemplo: ejecución de una instrucción

Suma de dos operandos: uno en memoria y otro en un registro del procesador



EJECUCIÓN DE UN PROGRAMA

- *En la ejecución normal del programa se ejecutan una a una las instrucciones, por orden.*
- La ejecución normal del programa puede interrumpirse si un dispositivo de E/S lo requiere, a través de una **señal de interrupción**.
- Una **INTERRUPCIÓN** es una petición de servicio de cualquier dispositivo de E/S al procesador. El procesador proporciona el servicio requerido con una **RUTINA DE INTERRUPCIÓN DE SERVICIO** adecuada.
- La interrupción podría cambiar el estado interno del procesador. Por eso, se guarda el contenido del PC, los registros generales y alguna información de control antes de dar servicio a la interrupción. Cuando se termina la interrupción, se restaura el estado del procesador y continúa ejecutándose el programa en el punto en que se había dejado.

5. Proceso Von Neumann

- Modelo Von Neumann: muy común (ej: PC).
- En Von Neumann ejecutar una instrucción puede tardar varios ciclos de reloj de la CPU.
- Su principal característica es cómo hace uso de la memoria: en Von Neumann datos e instrucciones se guardan en la misma memoria.

PROCESO DE VON NEUMANN

1. Se **accede a memoria** para buscar una instrucción.
 2. Se decodifica la instrucción. Se obtiene: 1) operación a realizar, 2) direcciones de los operandos.
 3. Se **accede a memoria** a buscar el operando 1.
 4. Se **accede a memoria** a buscar el operando 2.
 5. Se ejecuta la operación.
 6. Se **guarda en memoria** el resultado.
- Hay un mínimo de **4 accesos** a memoria.
 - Hay una única memoria para datos y programas y un solo bus con acceso a memoria -> se necesitan como mínimo 4 ciclos de reloj.

PROBLEMAS DEL ORDENADOR DE VON NEUMANN

- La separación entre el procesador (CPU) y la memoria produce lo que se conoce como el **cuello de botella de von Neumann**.
- La CPU está continuamente obligada a esperar que los datos necesarios sean transferidos hacia o desde la memoria del sistema.
- Esto produce una ralentización (disminución del rendimiento) debido a que si la velocidad de la CPU y tamaño de la memoria aumentan más rápido que el rendimiento entre ellos, el cuello de botella cada vez es mayor.

PROBLEMAS DEL ORDENADOR DE VON NEUMANN

SOLUCIONES:

1. Incorporar una caché entre la memoria principal y la CPU.
2. Proporcionar memorias independientes y accesos separados para instrucciones y datos (arquitectura Harvard).

En la actualidad la mayoría de los procesadores modernos se conectan al exterior de manera similar a la arquitectura Von Neumann, con un banco de memoria masivo único, pero internamente incluyen varios niveles de memoria caché con bancos separados en caché de programa y caché de datos, buscando un mejor desempeño sin perder la versatilidad.

3. Arquitecturas de computación paralela (sistemas con dos o más procesadores conectados entre sí que comparten la ejecución de una determinada tarea).

EJEMPLOS DE PROCESADORES BASADOS EN ARQUITECTURA DE VON NEUMANN

- Intel 80x86
- Intel® Xeon Phi™ Coprocessor (mirar el enlace en la comunidad sobre este procesador, apartado: **Evolution From Von Neumann Architecture to Intel® Xeon Phi™ Architecture**).

6. Proceso de Harvard

- Modelo Harvard: muy utilizado en aplicaciones específicas. Ej: **sistemas empotrados**.
- Modelo Harvard: recibe el nombre por el ordenador *Harvard Mark I*, desarrollado en esa universidad por Howard Aiken.
- En Von Neumann ejecutar una instrucción puede tardar varios ciclos de reloj de la CPU. En Harvard el objetivo es que una instrucción dure un ciclo (mucho más rápido).
- La principal diferencia está en el uso de la memoria: en Von Neumann datos e instrucciones se guardan en la misma memoria. En Harvard se guardan en distintas memorias.

COMPUTADORES

- DE ESCRITORIO:



- Hardware: procesador, memoria, E/S
- Software: Sistema Operativo, múltiples aplicaciones, podemos cambiar de aplicación.

- EMPOTRADOS:



- Hardware: procesador, memoria, E/S
- Software: puede tener Sistema Operativo, aplicación específica.

SISTEMAS EMPOTRADOS - ¿qué son?

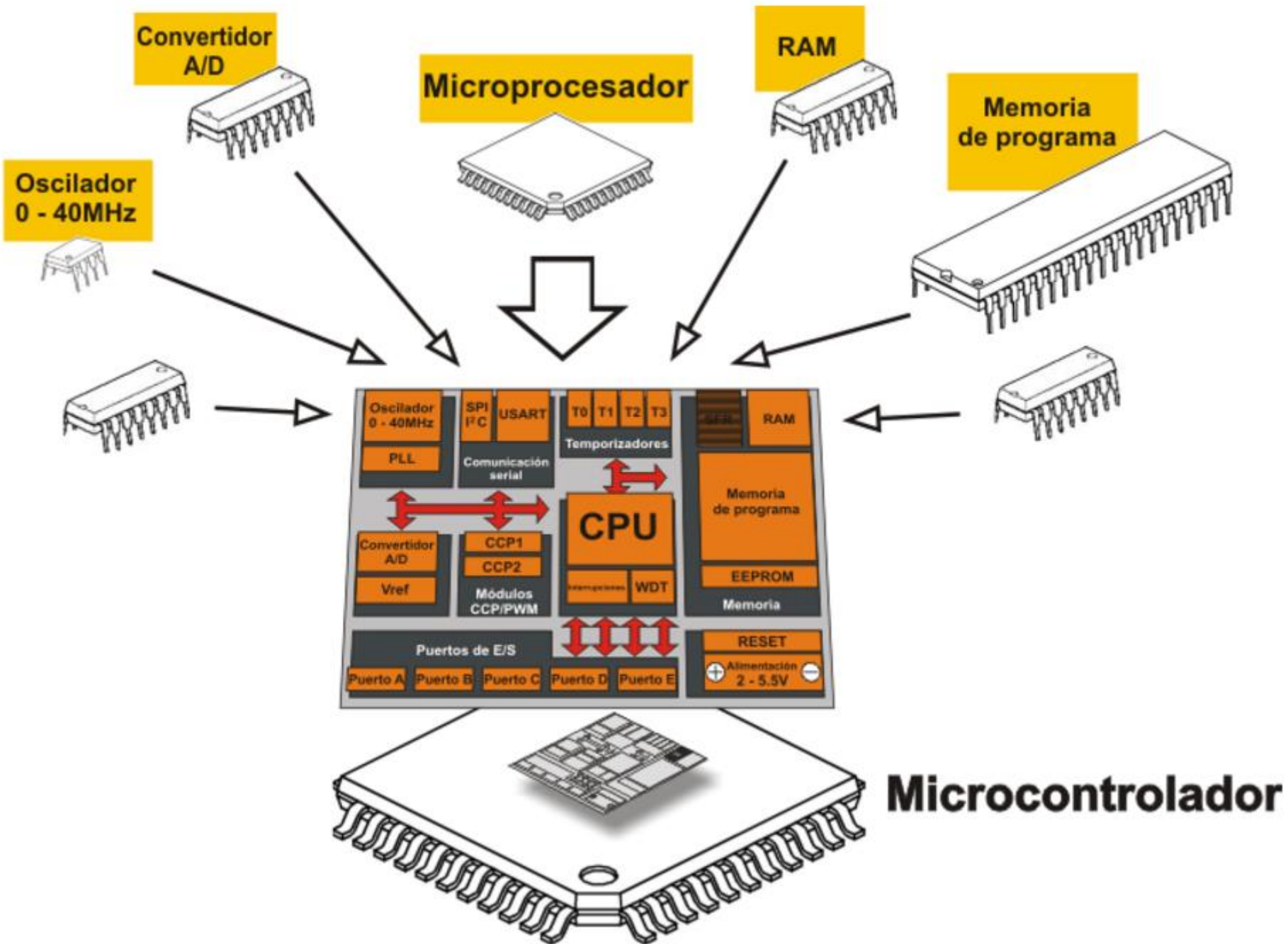
- Sistemas empotrados, o embebidos (*embedded systems*): sistema de computación diseñado para realizar una o algunas pocas funciones específicas en tiempo real. Al contrario de lo que ocurre con los ordenadores de propósito general (como por ejemplo una computadora personal o PC) que están diseñados para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para cubrir necesidades específicas.
- En un sistema embebido la mayoría de los componentes se encuentran incluidos en la placa base y no suelen tener el aspecto de lo que se suele asociar a una computadora.
- Ejemplos de sistemas embebidos: electrodomésticos (lavadoras), sistemas de comunicación (teléfonos), sistemas automotrices (frenos automáticos, encendido electrónico, control de velocidad), sistemas de vigilancia, cajas registradoras, cajeros automáticos, ascensores, dispositivos médicos (marcapasos), etc.

SISTEMAS EMPOTRADOS - características

- Por lo general, el usuario no los concibe como una computadora.
- El software tiene una misión fija.
- Relación muy estrecha entre software y hardware.
- Deben ser robustos.
- Bajo consumo de energía.
- Muy rápidos (funcionamiento en tiempo real).
- Autónomos.
- Computadores pequeños y económicos.

SISTEMAS EMPOTRADOS - tipos

- MICROCONTROLADOR: Microprocesador, memoria y dispositivos de E/S incluidos en un solo circuito integrado.
- DSP (*Digital Signal Processor*): microprocesador especializado en el procesamiento de señales digitales que funciona a tiempo real. Se suelen utilizar en el procesamiento de audio y video, y en comunicaciones digitales.
- CIRCUITOS INTEGRADOS ESPECÍFICOS PARA APLICACIÓN (ASIC): circuitos integrados personalizados para una aplicación en particular, como puede ser el control de un teléfono celular.
- FPGA (*Field Programmable Gate Array*): dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada mediante un lenguaje de descripción especializado.



PROCESO DE HARVARD

- Hay distintas memorias para 1) datos y 2) instrucciones, o 2) instrucciones + datos (“Harvard modificado”).

1. Se **accede a memoria 1** para buscar una instrucción.
 2. Se decodifica la instrucción. Se obtiene: 1) operación a realizar, 2) direcciones de los operandos.
 3. Se **accede a memoria 1** a buscar el operando 1.
 4. Se **accede a memoria 2** a buscar el operando 2.
 5. Se **ejecuta** la operación.
 6. Se **guarda en memoria 2** el resultado.
- Hay también 4 accesos a memoria. Pero **los pasos 1 y 6 y los 3 y 4 se pueden hacer a la vez**. Sólo se necesitan dos ciclos de reloj.

PROCESO DE HARVARD

- El modelo **Harvard** consigue ser más rápido que el modelo de Von Neumann.
- La mayor complejidad introducida por el modelo Harvard (duplicación de memoria, de buses) sólo compensa cuando el flujo de instrucciones y de datos es más o menos constante.
- Por eso no es ampliamente utilizada en ordenadores de propósito general.
- Hoy en día, con la masiva utilización de procesadores empuetrados en dispositivos móviles y dada la necesidad de una gran velocidad de procesamiento, la arquitectura de Harvard con modificaciones está en la base de la mayoría de los procesadores que hay en el mercado.

EJEMPLOS DE PROCESADORES DE TI (TEXAS INSTRUMENTS) BASADOS EN ARQUITECTURA HARVARD

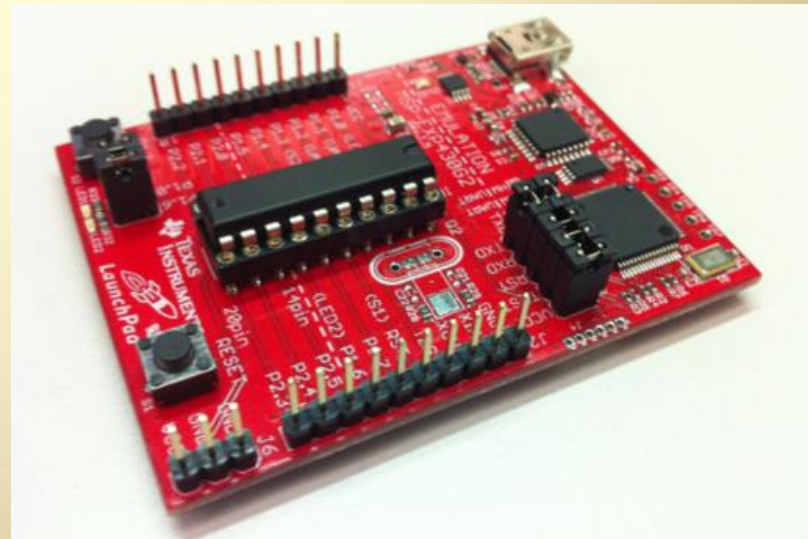
- **DSP**

- TMS320C67x
- TMS320F2837xD
- TMS320C50



- **Microcontroladores**

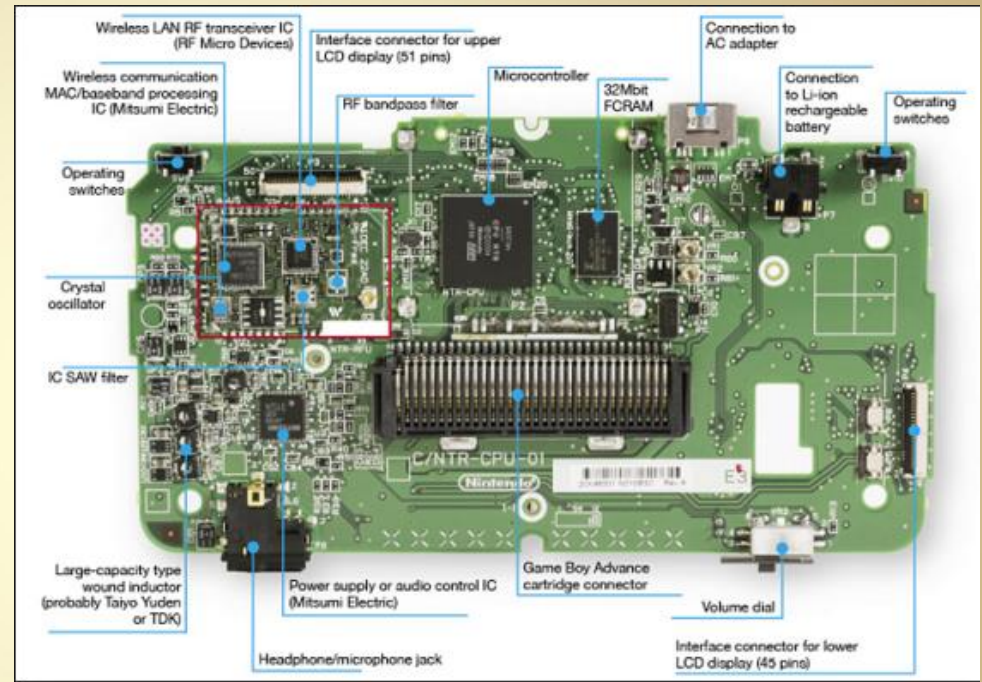
- MSP430



EJEMPLOS DE PROCESADORES DE ARM (ARM Holdings) BASADOS EN ARQUITECTURA HARVARD

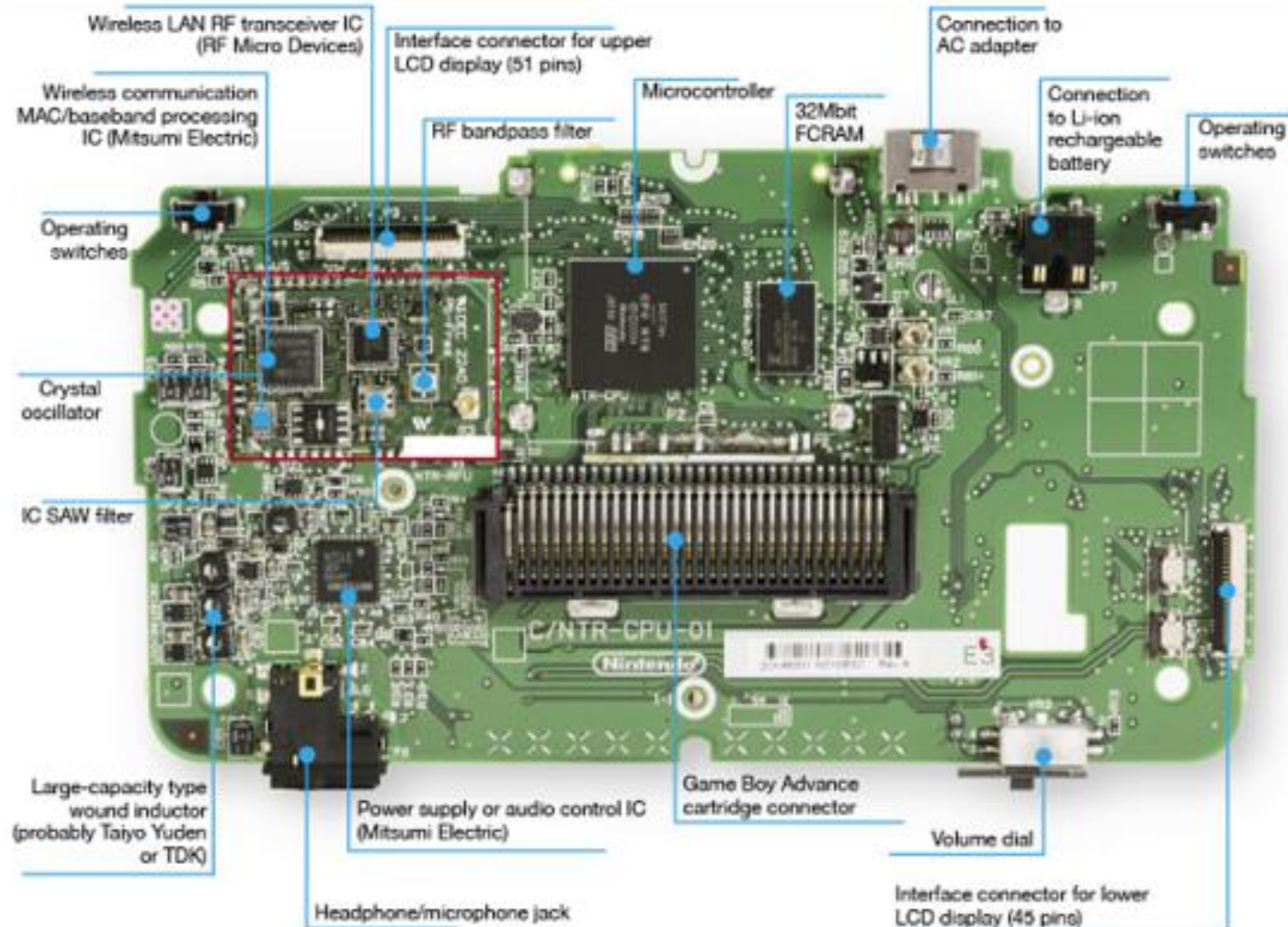
- Procesadores ARM

- Cortex-A8
- Cortex-M1
- Cortex-M3
- Cortex-M7
- Cortex-R5
- ARM920T
- ARM922T
- ARM946E-S



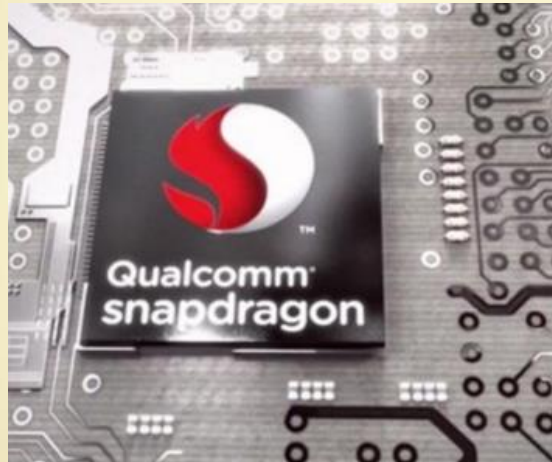
Nintendo DS – con un ARM946E-S

- Los procesadores desarrollados por esta compañía son utilizados por muchas otras: Apple, LG, Microsoft, Nintendo, Nokia, Sony, Yamaha, Texas Instruments, Qualcomm, Samsung, etc.



EJEMPLOS DE PROCESADORES QUALCOMM (Qualcomm Technologies) BASADOS EN ARQUITECTURA HARVARD

- Qualcomm Snapdragon 805 MSM8084
- Qualcomm Snapdragon 808 MSM8992



- Esta compañía americana está especializada en chips para telefonía móvil (Android). Es uno de los principales suministradores de procesadores para smartphones (Snapdragon).

EJEMPLOS DE PROCESADORES XILINX (Xilinx, Inc.) BASADOS EN ARQUITECTURA HARVARD

- Microcontrolador: MicroBlaze
- Microcontrolador: PIC165X
- Fast RISC Microcontroller (DFPIC165X)



- Esta compañía de tecnología americana es distribuidora de dispositivos lógicos programables. Es reconocida por inventar los FPGA. Actualmente abastece el 51% de todo el mercado de la lógica programable.

EJEMPLOS DE PROCESADORES SAMSUNG BASADOS EN ARQUITECTURA HARVARD

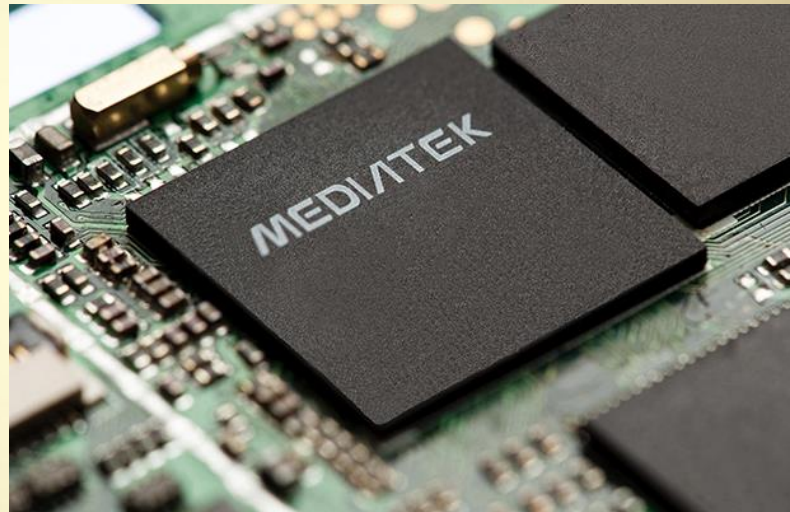
- Samsung MX5S (ARM v7)
- Samsung Exynos 5 Octa 5800 (ARM v7)
- Samsung S3C2410 SOC
- calmRISC32



- Samsung diseña y fabrica los procesadores Exynos, rivales en el mercado de los Qualcomm Snapdragon.

EJEMPLOS DE PROCESADORES MEDIATEK (MediaTek, Inc. – MTK) BASADOS EN ARQUITECTURA HARVARD

- MediaTek MT8125
- Mediatek MT6580



- Esta empresa china es la gran rival de la americana Qualcomm en el desarrollo de chips para teléfonos móviles Android.

EJEMPLOS DE PROCESADORES SPREADTRUM (Spreadtrum Communications, Inc.) BASADOS EN ARQUITECTURA HARVARD

- Spreadtrum SC7730A
- Spreadtrum SC8830



- Spreadtrum Communications es una compañía china proveedora de chips de SoC (System-on-Chip) para Smartphones, asociada a Intel, rival de Mediatek.

7. Conclusiones

- En Von Neumann ejecutar una instrucción puede tardar varios ciclos de reloj de la CPU. En Harvard el objetivo es que una instrucción dure un ciclo (mucho más rápido).
- La principal diferencia está en el uso de la memoria: en Von Neumann datos e instrucciones se guardan en la misma memoria. En Harvard se guardan en distintas memorias.
- **Modelo Von Neumann**: Utilizado en procesadores de propósito general (ordenadores personales).
- **Modelo Harvard**: Aunque antiguamente su uso era muy pequeño, hoy en día es la base de casi todos los procesadores empotrados.

BIBLIOGRAFÍA RECOMENDADA:

“Organización de computadores”. Hamacher, tema 1

“Organización y arquitectura de computadores”. Stallings, tema 2