

Clase 1 *****

DESCRIPCIÓN DE LA ASIGNATURA

- El objeto de la asignatura comprende el estudio de cada uno de los componentes de un aparato computador, utilizando técnicas de reparación, mantenimiento de sistemas y su montaje, en forma individual o en red de comunicaciones.

OBJETIVOS GENERALES

- a. Identificar los componentes internos de un computador.
- b. Conectar correctamente los periféricos de un computador.
- c. Identificar las posibles fallas debido a una conexión de los periféricos.
- d. Aplicar técnicas de mantenimiento a un computador y sus periféricos.

Arquitectura de computadoras

Introducción

Aplicación: Ofimática (MS-Office, Contapplus, D-Base)
Comunicaciones (Netscape , Explorer , Mail)
Diseño (AutoCAD, ...) , Multimedia , Juegos , etc.

Lenguaje de alto nivel: FOR, WHILE, REPEAT, PROCEDURE, ...
PASCAL, FORTRAN, C, COBOL, BASIC, ...
MODULA, C++, JAVA, ...

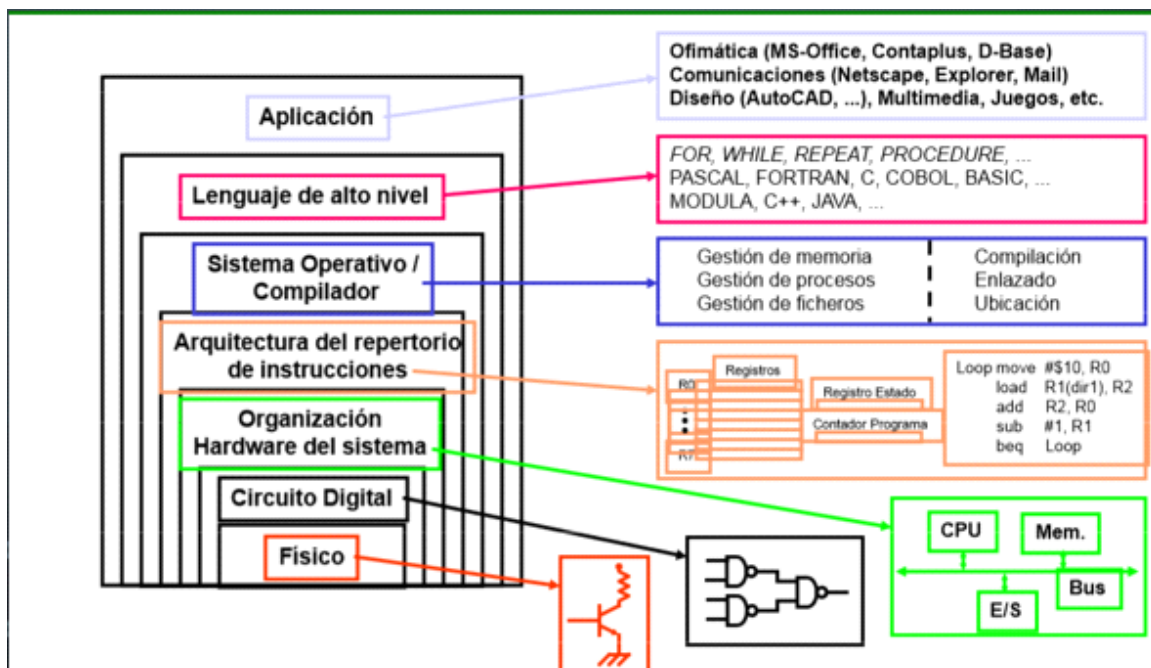
Sistema Operativo/Compilador: Gestión de memoria	Compilación
Gestión de procesos	Enlazado
Gestión de ficheros	Ubicación

Arquitectura del repertorio de instrucciones:

Organización Hardware del sistema: CPU, Memoria, E/S, Bus.

Circuito Digital:

Físico:



¿Dónde se estudia?

Aplicación: Casa, auto-aprendizaje en ratos libres, Academias de informática/ofimática.

Lenguaje de alto nivel: Fundamentos de Programación
Programación, EDI

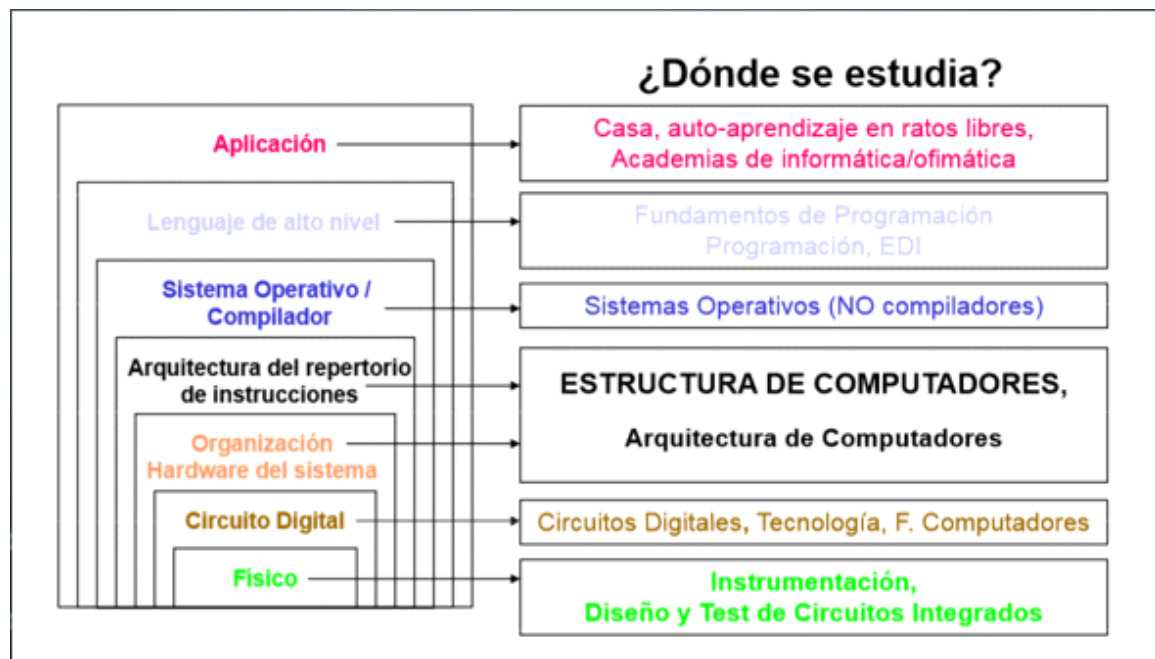
Sistema Operativo/Compilador: Sistemas Operativos (NO compiladores)

Arquitectura del repertorio de instrucciones: ESTRUCTURA DE COMPUTADORES, Arquitectura de Computadores

Organización Hardware del sistema: ESTRUCTURA DE COMPUTADORES, Arquitectura de Computadores.

Circuito Digital: Circuitos Digitales, Tecnología, F. Computadores.

Físico: Instrumentación, Diseño y Test de Circuitos Integrados.



Informática

- Ciencia que estudia el tratamiento de la información por medio de máquinas automáticas.
- Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores.
- Campo de conocimiento que abarca todos los aspectos de diseño y uso de ordenadores .

Computadora

- Máquina capaz de aceptar unos datos de entrada, efectuar con ellos operaciones lógicas y aritméticas, y proporcionar los datos resultantes a través de un medio de salida; todo ello sin la intervención de un operador humano y bajo el control de un programa de instrucciones previamente almacenado en el ordenador.

- **vs. Calculadora**

- Máquina capaz de efectuar operaciones aritméticas bajo el control directo del usuario:
 - No realiza operaciones de tipo lógico
 - No enlaza automáticamente las operaciones que realiza

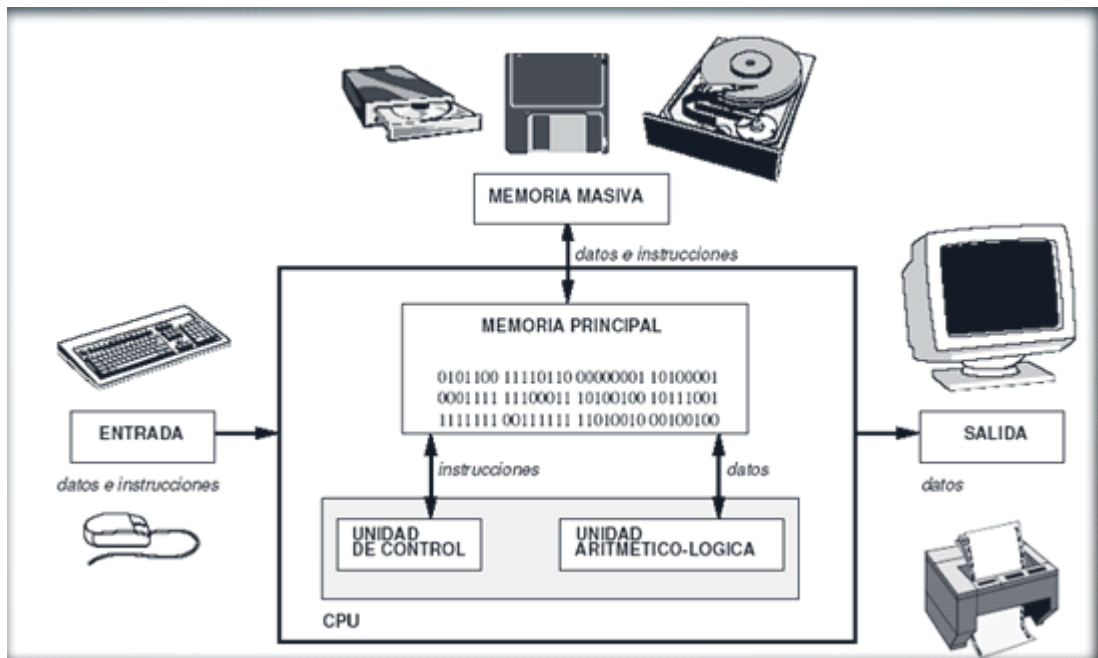
- **Ejemplos de operaciones aritméticas y lógicas**

Estructura funcional

de un ordenador:

Arquitectura von

Neumann



IDEA CLAVE: El programa se almacena en memoria junto con los datos.

Unidades de entrada y salida:

- Dispositivos por medio de los cuales se introducen datos e instrucciones en el ordenador.
- Dispositivos por donde se obtienen los resultados de los programas ejecutados por el ordenador.


○ Memoria:

- Donde se almacenan, por un corto o largo período de tiempo, tanto los datos como las instrucciones.

Memoria principal/central/interna

- La que trabaja a mayor velocidad
 - RAM [Random Access Memory]: De lectura y escritura, suele ser volátil.
 - ROM [Read Only Memory]: Sólo lectura (no se puede modificar), permanente.
- ¿Qué debe suceder para que un programa se ejecute?
- Para que un programa se ejecute, **debe estar cargado (almacenado) en la memoria principal**
- ¿Cómo se accede?
- Se estructura en posiciones de un determinado número de bits (palabras de memoria) **a las que se accede por su dirección.**

Dirección	Contenido
0	3745
1	2356
2	3725
3	4832
4	2437
5	4326
⋮	⋮
255	3456

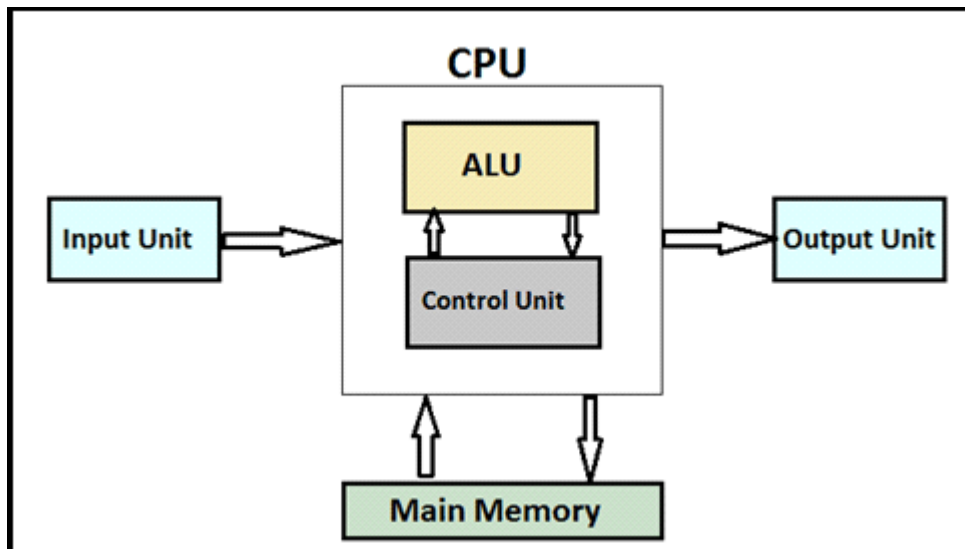


Memoria auxiliar/secundaria/externa/masiva

- Más lenta, pero de mayor capacidad que la memoria principal
- ¿Adónde se suelen almacenar los datos y programas y para qué?
- Los datos y programas se suelen almacenar **en la memoria auxiliar** para que, **cuando se ejecute varias veces un programa o se utilicen repetidamente unos datos, no sea necesario introducirlos de nuevo.**

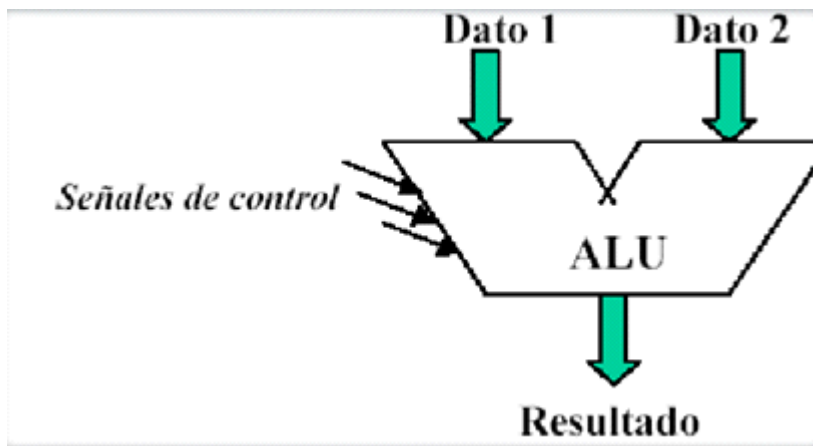
CPU (CentralProcessing Unit)

CPU = ALU + UC



Unidad aritmético-lógica (UAL o ALU)

- o Unidad encargada de efectuar las operaciones aritméticas y lógicas.



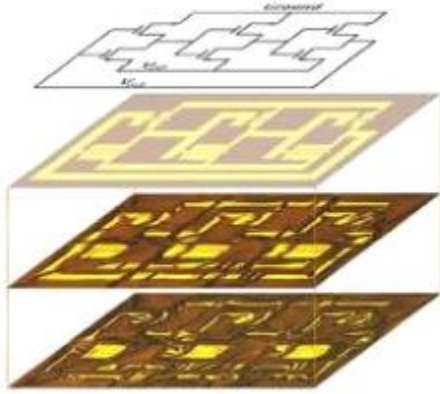
Unidad de control (UC)

- o ¿Qué hace la unidad de control?

- Detecta señales de estado procedentes de las distintas partes del ordenador y genera señales de control dirigidas a todas las unidades para controlar el funcionamiento de la máquina.
 - Capta de la memoria principal las instrucciones del programa que ejecuta el ordenador, las descodifica y las ejecuta una a una.
- ¿Qué contiene la unidad de control? ¿Qué hace el reloj?
 - Contiene un reloj que sincroniza todas las operaciones elementales involucradas en la ejecución de una instrucción.
 - ¿Qué determina la frecuencia del reloj?
 - La frecuencia del reloj (en Megahertzios, MHz, o Gigahertzios, GHz) determina, en parte, la velocidad de funcionamiento del ordenador.

Microprocesador:

CPU contenida en uno o varios circuitos integrados (chips)



Ejecución de un programa

Una vez cargado el programa en memoria, se le cede el control del ordenador:

1. Se lee una instrucción del programa.
2. La unidad de control decodifica la instrucción
3. La unidad de control envía las señales necesarias para ejecutar la instrucción:
 - a. Se leen los datos de entrada (1 ó 2)
 - b. Se efectúa una operación con ellos en la ALU (p.ej. suma)
 - c. Se almacena el resultado
4. Se determina cuál debe ser la siguiente instrucción que se debe ejecutar.
5. Se vuelve al paso 1.

Tipos de instrucciones:

Instrucciones de transferencia de datos:

Mover datos de un sitio a otro.

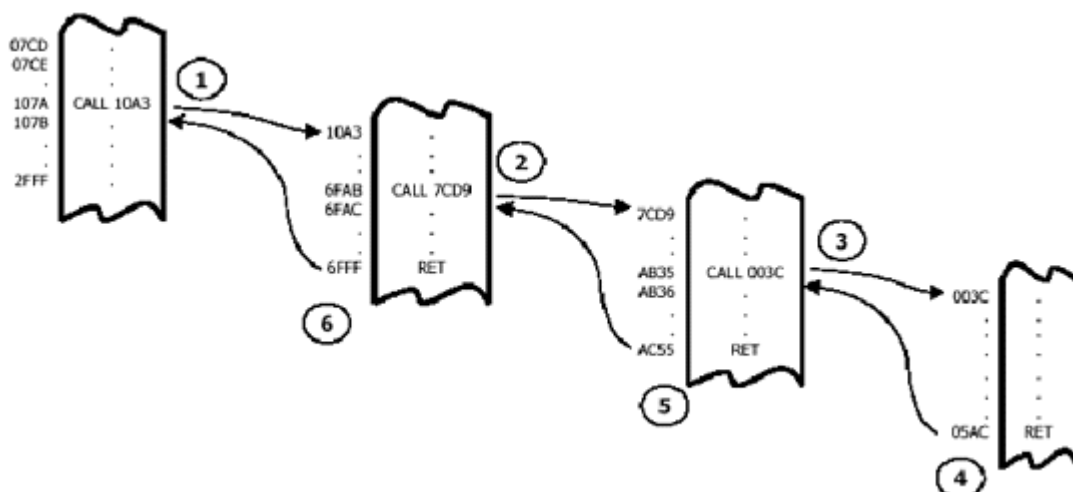
Instrucciones de tratamiento:

Realización de operaciones aritméticas y lógicas.

Instrucciones de bifurcación y saltos:

Alteración del orden secuencial de ejecución de las instrucciones.

ej.: Llamadas/retornos de subrutinas



Clase 3 *****

Funcionamiento de una Computadora.

Generalidades del Funcionamiento de una Computadora.

¿Basándose en qué y Cómo se han diseñado virtualmente las computadoras actuales?

R/

- Virtualmente las computadoras actuales se han diseñado basándose en los conceptos que desarrollo John Von Neumann, el cual posee 3 conceptos claves:

- o Los Datos y las instrucciones se almacenan en una sola memoria de lectura-escritura.
- o Los contenidos de esta memoria se direccionan indicando su posición, sin considerar el tipo de dato contenido en la misma.
- o La ejecución se produce siguiente una secuencia de instrucciones tras instrucciones.

¿Dónde se encuentra el éxito de este esquema?

R/

- Recordemos como eran las computadoras a un inicio.

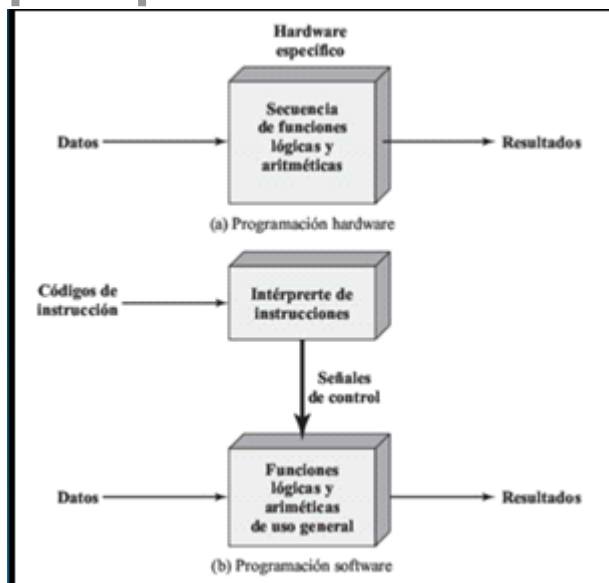


- “Si se desea realizar un cálculo concreto, es posible utilizar una configuración de componentes lógicos diseñada específicamente para dicho cálculo”

• Consideremos el siguiente escenario.

- “Se construye una configuración de uso general de funciones lógicas y aritméticas. Este hardware realizará funciones diferentes según las señales de control aplicadas. En el caso del hardware específico, el sistema acepta datos y produce resultados; con el hardware de uso general, el sistema acepta datos y señales de control y produce resultados. Así, en lugar de reconfigurar el hardware para cada nuevo programa, el programador simplemente necesita proporcionar un nuevo conjunto de señales de control.”

• Analicemos el concepto desde una perspectiva más visual.

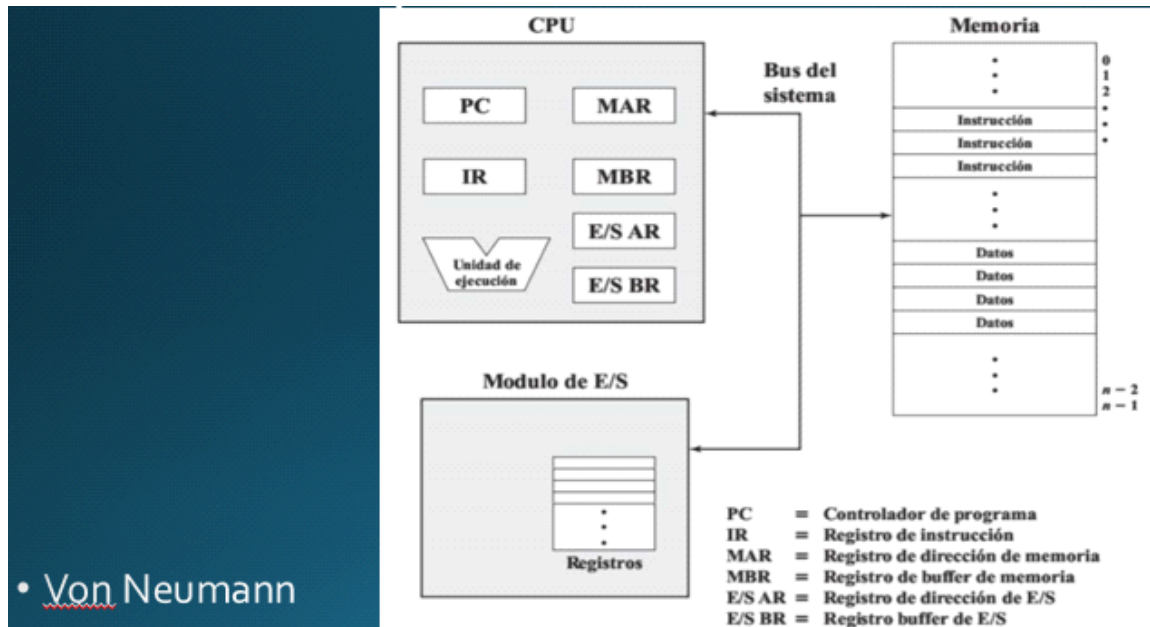


Componentes esenciales del sistema.

La figura anterior muestra dos componentes esenciales del sistema: **Un intérprete de instrucciones y un módulo de uso general para las funciones aritméticas y lógicas**. Estos dos elementos constituyen la CPU. Se requieren varios componentes adicionales para que el computador pueda

funcionar, estos son los que conocemos como:

- **Componentes de Entrada y Salida:** Permite la entrada de datos y traducirlos al formato de señales que utiliza el sistema.
- **Memoria Principal:** Un programa no ejecuta las sentencias bajo una secuencia, puede saltarse ciertas instrucciones, por ello existe la memoria principal para almacenar temporalmente las instrucciones como los datos.



¿De qué se encarga la CPU?

R/

La CPU se encarga del control. Intercambia datos con la memoria.

- Para ello, usualmente utiliza dos registros internos (en la CPU):
- Un Registro de Direcciones de Memoria (MAR, por sus siglas en inglés Memory Address Register), que especifica la dirección en memoria de la próxima lectura o escritura.
- Un Registro para Datos de Memoria (MBR, por sus siglas en inglés Memory Buffer Register), que contiene el dato que se va a escribir en memoria o donde se escribe el dato que se va a leer de memoria.
- Un Registro de Direcciones de E/S (E/S AR, por sus siglas en inglés E/S Address Register) especifica un dispositivo de E/S.
- Un Registro para Datos de E/S (E/S BR, por sus siglas en inglés E/S Buffer Register) Se utiliza para intercambiar datos entre un módulo de E/S y la CPU.

Ejecución de Programas

¿Cuál es la función básica que realiza una computadora?

R/

- La función básica que realiza una computadora es la ejecución de un programa, constituido por un conjunto de instrucciones almacenadas en memoria.

¿Se encarga de ejecutar las instrucciones especificadas en el programa?

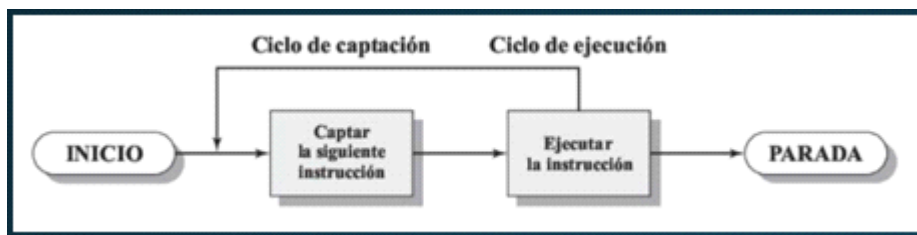
R/

- El procesador es precisamente el que se encarga de ejecutar las instrucciones especificadas en el programa.
- Los aspectos claves en la ejecución de un programa, que en su forma más simple consta de dos etapas: **El procesador lee (capta) la instrucción de memoria, y la ejecuta.** ¿En qué consiste la ejecución del programa? R/ **La ejecución del programa consiste en la repetición del proceso de captación de instrucción y ejecución de instrucción.**

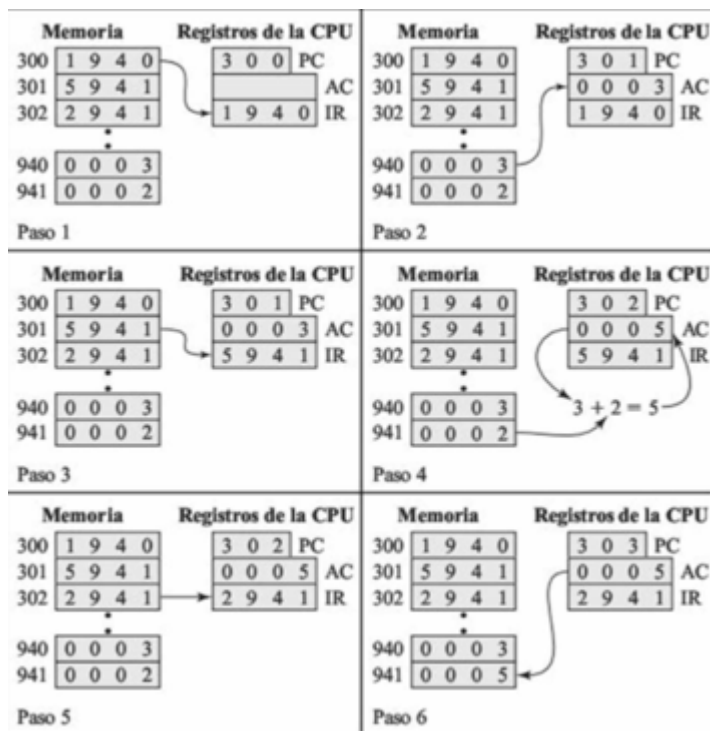
¿Cómo se denomina el procesamiento que requiere una instrucción?

R/

- **El procesamiento que requiere una instrucción se denomina ciclo de instrucción.** Se representa en la Figura utilizando la descripción simplificada de dos etapas explicada en la lámina anterior. ¿Cómo se denotan los dos pasos? R/ **Los dos pasos se denotan como ciclo de captación y ciclo de ejecución.**



- **Ejemplo:** Se ilustra la ejecución de una parte de un programa, mostrando las partes relevantes de la memoria y los registros de la CPU. El fragmento del programa suma el contenido de la palabra de memoria en la dirección 940 con el contenido de la palabra de la memoria en la dirección 941 y almacena el resultado en esta última posición.
- Para este ejemplo se requieren **3 Instrucciones**, que consumen 3 ciclos de captación y 3 de ejecución.



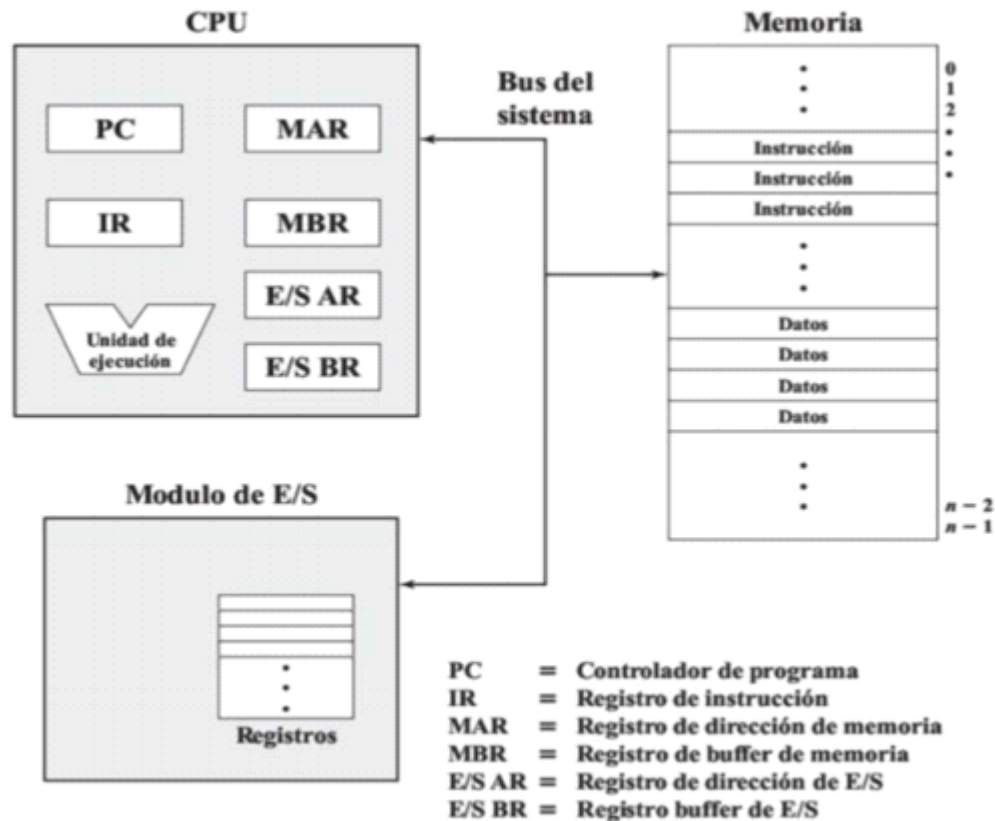
- Figura de Ejemplo

¿Cómo funciona?

1. El **contador de programa (PC)** contiene el valor 300, la dirección de la primera instrucción. Esta instrucción (el valor hexadecimal 1940) se carga en el registro de instrucción (IR). Observe que este proceso implicaría el uso del registro de dirección de memoria (MAR) y el registro de datos de memoria (MBR). Por simplicidad, se han ignorado estos registros intermedios.
2. Los primeros cuatro bits de IR (primer dígito hexadecimal) indican que el acumulador (AC) se va a cargar. Los restantes 12 bits (tres dígitos hexadecimales) especifican la dirección (940) que se va a cargar.
3. El registro PC se incrementa, y se capta la siguiente instrucción (5941) desde la dirección 301.
4. El contenido anterior de AC y el de la posición de memoria 941 se suman, y el resultado se almacena en AC.
5. El registro PC se incrementa, y se capta la siguiente instrucción (2941) desde la posición 302.
6. El contenido de AC se almacena en la posición 941.

Los Ciclos de Captación y Ejecución

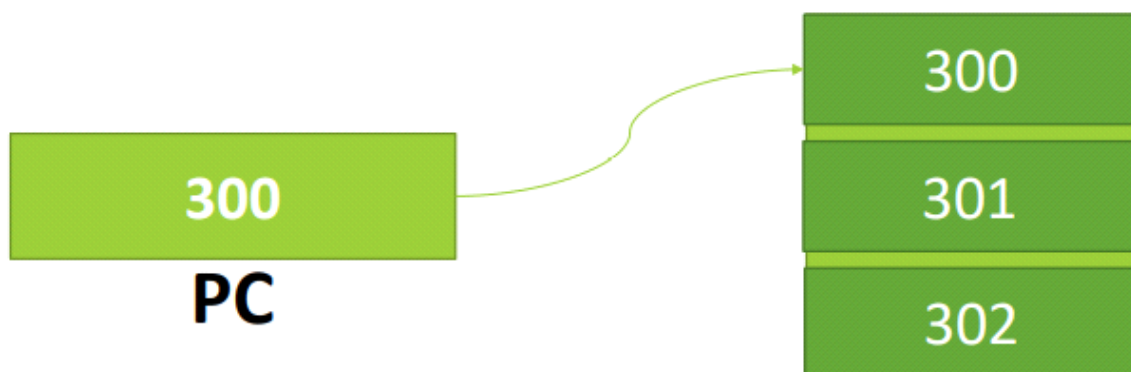
Al comienzo de cada ciclo de instrucción, la CPU capta una instrucción de memoria. En una CPU típica, se utiliza un registro llamado contador de programa (**PC, Program Counter**) para seguir los flujos de la instrucción que debe ejecutar a continuación.



A no ser que se indique otra cosa, la CPU siempre incrementa el PC de comprender cada instrucción, de forma que comprende la siguiente instrucción de la secuencia.

Ejemplo: Una computadora en que la instrucción ocupa una palabra de **memoria de 16 bits**. Se supone que el contador de programa almacena el valor **300**. El CPU captará la próxima instrucción almacenada en la posición **301, 302, 303**. Esta secuencia es posible alterarla.

“Lo Almacenado por el PC son direcciones de memoria, que apunta al IR”



La instrucción captada se almacena en un registro de la CPU conocido como **registro de instrucción**

(IR, Instruction Register). La **instrucción** se escribe utilizando un código binario que especifica la acción que debe realizar el CPU. El CPU interpreta la instrucción y lleva a cabo la acción requerida. En general, esta puede ser de cuatro tipos:

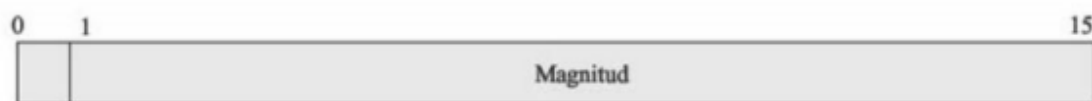
TIPOS:

- **Procesador-Memoria o Procesador Memoria:** Deben transferirse datos desde la CPU a la memoria, o desde la memoria a la CPU.
- **Procesador-E/S o Procesador E/S:** Deben transferirse datos a o desde el exterior mediante transferencias entre la CPU y un módulo de E/S.
- **Procesamiento de Datos:** El CPU ha de realizar alguna operación aritmética o lógica con los datos.
- **Control:** Una instrucción puede especificar que la secuencia de ejecución se altere (como la instrucción de salto LAS). Por ejemplo, El CPU capta una instrucción de la posición 149 que especifica que la siguiente instrucción debe captarse de la posición 182. El CPU registrará este hecho poniendo en el contador de programa 182. Así, en el próximo ciclo de captación, la instrucción se cargará desde la posición 182 en lugar de desde la posición 150.

Otro Ejemplo: Considere un ejemplo sencillo donde utilizamos una “maquina hipotética” que incluye las características enumeradas a continuación:



(a) Formato de instrucción



(b) Formato de enteros

Contador de programa (PC) = Dirección de instrucción
Registro de instrucción (IR) = Instrucción en ejecución
Acumulador (AC) = Almacenamiento temporal

(c) Registros internos de la CPU

0001 = Cargar AC desde memoria
0010 = Almacenar AC en memoria
0101 = Sumar a AC un dato de memoria

(d) Lista parcial de Codops («códigos de operación»)

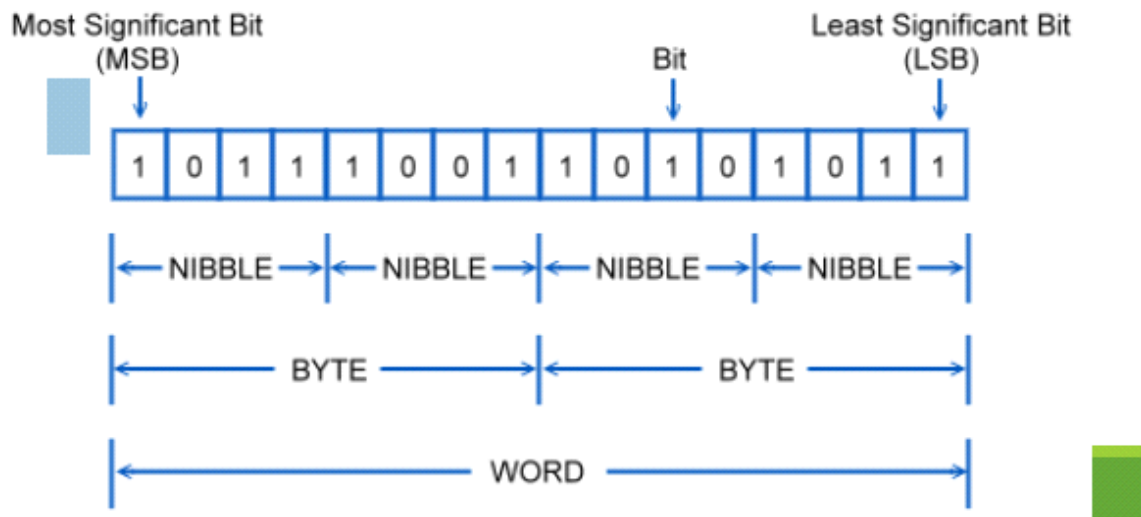
¿Único registro de datos del Procesador?

R/Acumulador AC

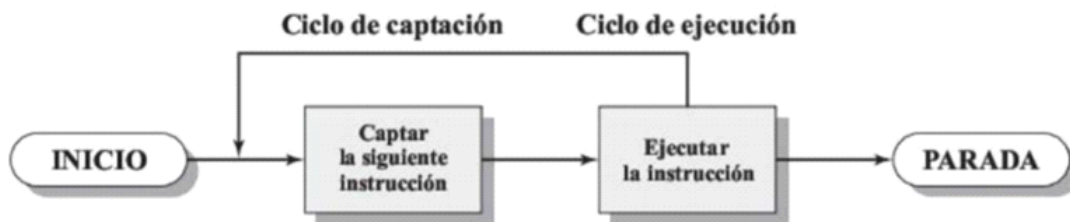
¿Hasta cuantas palabras de memoria se pueden direccionar?

R/ $2^{12} = 4096$ (4K) palabras de memoria.

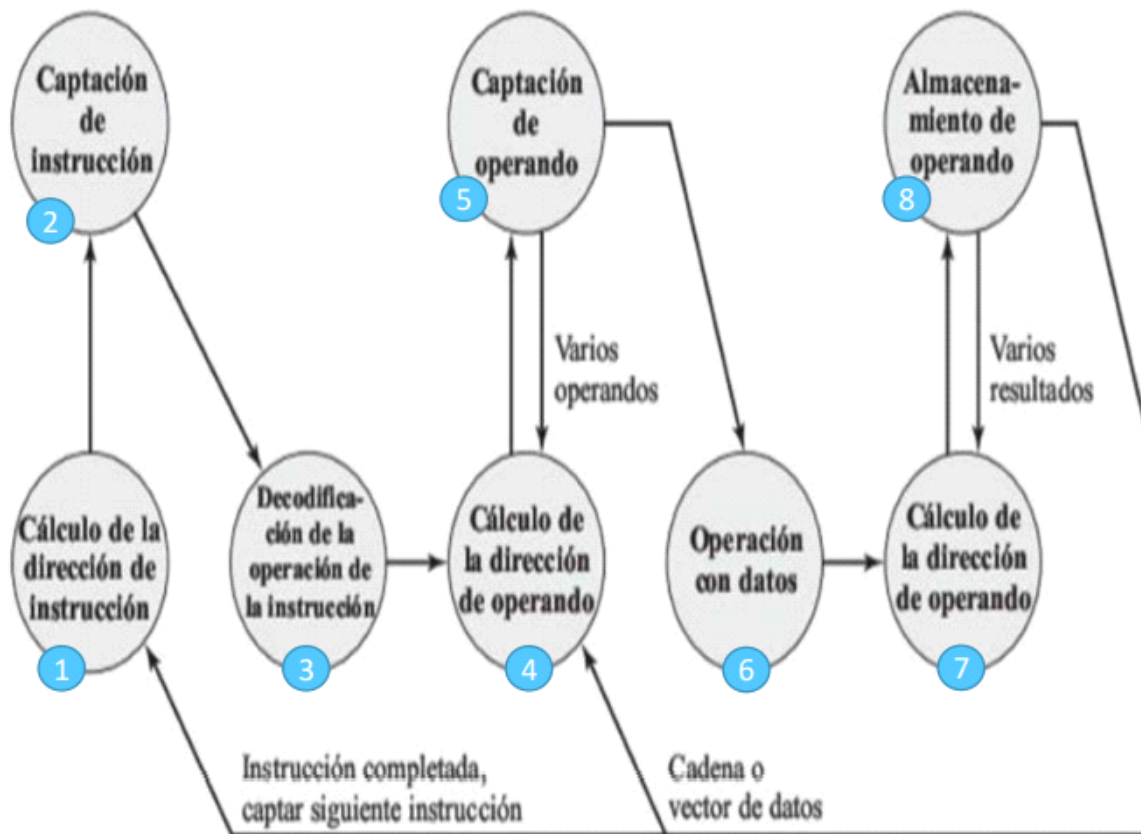
El procesador posee un único registro de datos **llamado acumulador (AC)**. Tanto las instrucciones como los datos son de 16 bits. Así, es conveniente organizar la memoria utilizando posiciones de 16 bits, o **palabras**. El formato de instrucción indica que puede haber **$2^4 = 16$ códigos de operación (codops) diferentes**, y se pueden direccionar directamente hasta **$2^{12} = 4096$ (4K) palabras de memoria**.



Si los Ciclos de Instrucciones fueran en particular solo para la captación y ejecución de Instrucciones o de referenciar memoria podríamos mantener nuestro esquema inicial intacto:



Pero nuestras computadoras en vez de referenciar solamente memoria, o una instrucción puede especificar una operación de E/S.



Para un ciclo de instrucción dado, algunos estados pueden no darse y otros pueden visitarse más de una vez. Los estados se describen a continuación:

1. Cálculo de la dirección de la instrucción (IAC, Instruction Address Calculation): Determina la dirección de la siguiente instrucción a ejecutar. Normalmente, esto implica añadir un número fijo a la dirección de la instrucción previa.

Por ejemplo, si las instrucciones tienen un tamaño de 16 bits y la memoria se organiza en palabras de 16 bits, se suma 1 a la dirección previa. En cambio, si la memoria se organiza en bytes (8 bits) direccionables individualmente, entonces hay que sumar 2 a la dirección previa.

2. Captación de instrucción (if, Instruction Fetch): El CPU lee la instrucción desde su posición en memoria.

3. Decodificación de la operación indicada en la instrucción (IOD, Instruction Operation Decoding): analiza la instrucción para determinar el tipo de operación a realizar y el (los) operando(s) a utilizar o y el operando a utilizar o y los operandos a utilizar.

4. Cálculo de la dirección del operando (OAC, Operand Address Calculation): Si la instrucción implica una referencia a un operando en memoria o disponible mediante E/S, determina la dirección del operando.

5. Captación de operando (OF, Operand Fetch): Capta el operando desde memoria o se lee desde el dispositivo de E/S.

6. Operación con los datos (DO, Data Operation): Realiza la operación indicada en la instrucción.

7. Realizamos lo descrito en el PASO 4....

8. Almacenamiento de operando (OS, Operand Store): Escribe el resultado en memoria o lo saca a través de un dispositivo de E/S.

¿Por qué es importante que conozcamos cómo funcionan los ciclos de captación y Ejecución?

Es importante ya que ahora sabemos que existe la posibilidad de que en los ciclos de instrucción, pueda existir la interacción de los elementos de E/S. Y para ello es importante además que conozcamos el tema de: **INTERRUPCIONES**

Clase 5 *****

Principio de Diseño de Repertorio de Instrucciones

Clasificación de las Arquitecturas a nivel lenguaje máquina.

- Realizaremos una breve introducción a las arquitecturas a nivel lenguaje máquina, la parte de la maquina visible al programador, el cual está compuesta por los **Repertorios de Instrucciones**.
- Los repertorios de instrucciones se pueden clasificar en general siguiendo las cinco dimensiones descritas en la imagen, que están ordenadas aproximadamente por el papel que juegan en diferenciar los repertorios de instrucciones.

Clasificación de las Arquitecturas a nivel lenguaje máquina.

Almacenamiento de operandos en la CPU	Además de en memoria ¿dónde se encuentran los operandos?
Número de operandos explícitos por instrucción	¿Cuántos operandos son designados explícitamente en una instrucción típica?
Posición del operando	¿Puede cualquier operando de una instrucción de la ALU estar localizado en memoria o deben algunos o todos los operandos estar en la memoria interna de la CPU? Si un operando está localizado en memoria, ¿cómo se especifica la posición de memoria?
Operaciones	¿Qué operaciones se proporcionan en el repertorio de instrucciones?
Tipo y tamaño de operandos	¿Cuál es el tipo y tamaño de cada operando y cómo se especifica?

Almacenamiento de operandos en la CPU	Además de en memoria ¿dónde se encuentran los operandos?
Número de operandos explícitos por instrucción	¿Cuántos operandos son designados explícitamente en una instrucción típica?
Posición del Operando	¿Puede cualquier operando de una instrucción de la ALU estar localizado en memoria o deben algunos o todos los operandos estar en la memoria interna de la CPU? Si un operando está localizado en memoria, ¿cómo se especifica la posición de memoria?
Operaciones	¿Qué operaciones se proporcionan en el repertorio de instrucciones?
Tipo y tamaño de	¿Cuál es el tipo y tamaño de cada operando y cómo se especifica?

- El tipo de almacenamiento proporcionado para que los operandos estén en la CPU, en contraposición a que estén en memoria, es el factor distintivo más importante entre las arquitecturas a nivel lenguaje máquina. **(Todas las arquitecturas conocidas tienen algún almacenamiento temporal en la CPU).**
- El tipo de almacenamiento de operandos de la CPU a veces dicta el número de operandos explícitamente nombrados en una instrucción. En una clase de máquinas, el número de operandos explícitos puede variar.

Alternativas de Almacenar Operando en el CPU.

Almacena- miento temporal proporcionado	Ejemplos	Operandos explícitos por instrucción ALU	Destino para resultados	Procedimiento para acceder a operandos explícitos
Pila	B5500, HP 3000/70	0	Pila	Introducir y sacar de la pila
Acumulador	PDP-8 Motorola 6809	1	Acumulador	Cargar/ almacenar acumulador
Conjunto de registros	IBM 360, DEC VAX	2 ó 3	Registros o memoria	Cargar/ almacenar registros o memoria

Almacenamiento temporal proporcionado	Ejemplos	Operandos explícitos por instrucción ALU	Destino para resultados	Procedimiento para acceder a operandos explícitos
Pila	B5500, HP 3000/70	0	Pila	Introducir y sacar de la pila
Acumulador	PDP-8 Motorola 6809	1	Acumulador	Cargar/almacenar acumulador
Conjunto de registros	IBM 360, DEC VAX	2 ó 3	Registros o memoria	Cargar/almacenar registros o memoria

- Las elecciones principales son una pila, un acumulador o un conjunto de registros. Los operandos pueden ser nombrados **explícita o implícitamente**:

- **Los operandos en una arquitectura de pila** están implícitamente en el tope de la pila;

Arquitectura acumulador

- **En una arquitectura de acumulador** un operando está implícitamente en el acumulador.

- **Las arquitecturas de registros de propósito general** tienen solamente operandos explícitos en registros o en posiciones de memoria. Dependiendo de la arquitectura, los operandos explícitos para una operación pueden ser accedidos directamente desde memoria o puede ser necesario cargarlos primero en el almacenamiento temporal, dependiendo de la clase de instrucción y elección de la instrucción específica

- **Ejemplo:** Mostrar cómo sería la secuencia de código $C = A + B$ en estas tres clases de repertorios de

instrucciones.

Pila	Acumulador	Registro
PUSH A	LOAD A	LOAD R1,A
PUSH B	ADD B	ADD R1,B
ADD	STORE C	STORE C, R1
POP C		

Pila	Acumulador	Registro
PUSH A	LOAD A	LOAD R1,A
PUSH B	ADD B	ADD R1,B
ADD	STORE C	STORE C,R1
POP C		

• **Ventajas y Desventajas:**

Tipo de máquina	Ventajas	Desventajas
Pila	Modelo sencillo para evaluación de expresiones (polaca inversa). Instrucciones cortas pueden dar una buena densidad de código.	A una pila no se puede acceder aleatoriamente. Esta limitación hace difícil generar código eficiente. También dificulta una implementación eficiente, ya que la pila llega a ser un cuello de botella.
Acumulador	Minimiza los estados internos de la máquina. Instrucciones cortas.	Como el acumulador es solamente almacenamiento temporal, el tráfico de memoria es el más alto en esta aproximación.
Registro	Modelo más general para generación de código.	Todos los operandos deben ser nombrados, conduciendo a instrucciones más largas.

Almacenando Operando en Memoria .

• Los registros permiten una ordenación más flexible que las pilas o acumuladores, a la hora de evaluar las expresiones. Por ejemplo, en una máquina de registros, la expresión **(A .B) - (C . D) - (E . F)** puede evaluarse haciendo las multiplicaciones en cualquier orden, lo que puede ser más eficiente debido a la posición de los operandos o a causa de las posibilidades de la segmentación.

• Pero en una máquina de pilas la expresión debe evaluarse de izquierda a derecha, a menos que se realicen operaciones especiales o intercambios de las posiciones de la pila.

DOS CARACTERÍSTICAS IMPORTANTES DE LOS REPERTORIOS DE INSTRUCCIONES:

¿Ambas están relacionadas con la naturaleza de los operandos?

• Hay dos características importantes de los repertorios de instrucciones que dividen las arquitecturas de **registros de propósito general o GPR**. Ambas características están relacionadas con la naturaleza de los operandos para una instrucción lógica o aritmética, o instrucción de la ALU:

Primer Característica

¿A qué se refiere?

- La primera se refiere al número de operandos (dos o tres) que pueden tener las instrucciones de la ALU. En el formato de tres operandos, la instrucción contiene un resultado y dos operandos fuente.

- En el formato de dos operandos, uno de los operandos es fuente y destino para la operación

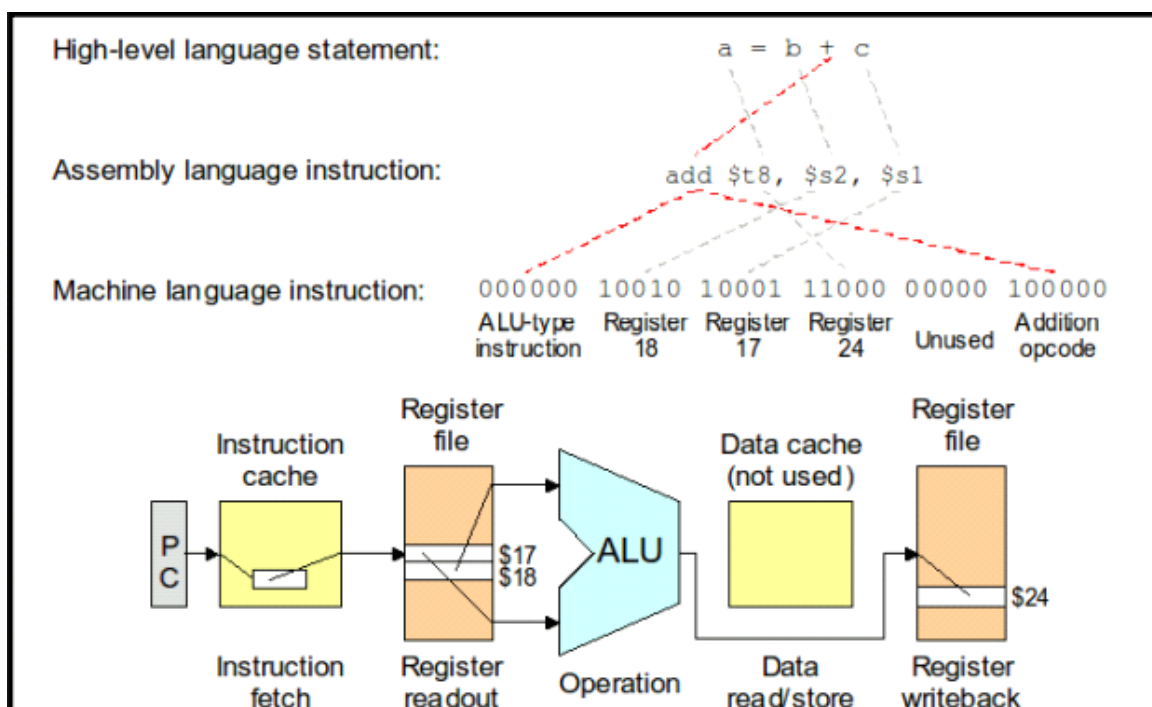
- La segunda característica de las arquitecturas GPR está relacionada con el número de operandos que se pueden direccionar en memoria en las instrucciones de la ALU. Este número puede variar de cero a tres. Todas las combinaciones posibles de estos dos atributos se muestran en la figura que se mostrar a continuación, con ejemplos de máquinas. Aunque hay siete posibles combinaciones tres sirven para clasificar aproximadamente todas las máquinas existentes: **registro-registro (también llamadas de carga/almacenamiento), registro-memoria y memoria-memoria.**

Número de direcciones de memoria por instrucción típica de la ALU	Máximo número de operandos permitidos por instrucción típica de la ALU	Ejemplos
0	2 3	IBM RT-PC SPARC, MIPS, HP Precision Architecture
1	2 3	PDP-10, Motorola 68000, IBM 360 Parte de IBM 360 (instrucciones RS)
2	2 3	PDP-11, National 32x32, parte de IBM 360 (instrucciones SS)
3	3	VAX (también tiene formatos de dos operandos)

Número de direcciones de memoria por instrucción típica de la ALU	Máximo número de operandos permitidos por instrucción típica de la ALU	Ejemplos
0	2 3	IBM RT-PC SPARC, MIPS, HP Precision Architecture
1	2 3	PP-10, Motorola 68000, IBM 360 Parte de IBM 3600 (instrucciones RS)
2	2 3	PDP-11, National 32x32, parte de IBM 360 (instrucciones SS)
3	3	VAX (también tiene formatos de dos operandos)

Tipo	Ventajas	Desventajas
Registro-registro (0,3)	Codificación simple de instrucciones de longitud fija. Modelo simple de generación de código. Las instrucciones emplean números de ciclos similares para ejecutarse (ver Capítulo 6).	Recuento más alto de instrucciones que las arquitecturas con referencias a memoria en las instrucciones. Algunas instrucciones son cortas y la codificación de bits puede ser excesiva.
Registro-memoria (1,2)	Los datos pueden ser accedidos sin cargarlos primero. El formato de instrucción tiende a ser fácil para codificar y obtener buena densidad.	Los operandos no son equivalentes, ya que en una operación binaria se destruye un operando fuente. Codificar un número de registro y una dirección de memoria en cada instrucción puede restringir el número de registros. Los ciclos por instrucción varían por la posición de operando.
Memoria-memoria (3,3)	Más compacta. No emplean registros para temporales.	Gran variación en el tamaño de las instrucciones, especialmente en instrucciones de tres operandos. Además gran variación en el trabajo por instrucción. Los accesos a memoria crean cuellos de botella en memoria.

Tipo	Ventajas	Desventajas
Registro-registro	Codificación simple de instrucciones de longitud fija. Modelo simple de generación de código. Las instrucciones emplean números de ciclos similares para ejecutarse (ver Capítulo 6).	Recuento más alto de instrucciones que las arquitecturas con referencias a memoria en las instrucciones. Algunas instrucciones son cortas y la codificación de bits puede ser excesiva.
Registro-memoria (1,2)	Los datos pueden ser accedidos sin cargarlos primero. El formato de instrucción tiende a ser fácil para codificar y obtener buena densidad.	Los operandos no son equivalentes, ya que en una operación binaria se destruye un operando fuente. Codificar un número de registro y una dirección de memoria en cada instrucción puede restringir el número de registros. Los ciclos por instrucción varían por la posición de operando.
Memoria-memoria (3,3)	Más compacta. No emplean registros para temporales.	Gran variación en el tamaño de las instrucciones, especialmente en instrucciones de tres operandos. Además gran variación en el trabajo por instrucción. Los accesos a memoria crean cuellos de botella en memoria.



DIRECCIONAMIENTO DE MEMORIA

¿Arquitectura registro-registro llamada también?

- Independientemente que una arquitectura sea registro-registro (**también llamada de carga/almacenamiento**) o permita que cualquier operando sea una referencia a memoria, debe definir cuántas direcciones de memoria son interpretadas y cómo se especifican.

- Las medidas presentadas aquí son muy, pero no completamente, independientes de la máquina. En algunos casos, las medidas están afectadas significativamente por la tecnología del compilador. Estas medidas se han realizado utilizando un compilador optimizador, ya que la tecnología de compiladores está jugando un papel creciente. Las medidas probablemente reflejarán lo que veremos en el futuro antes que lo que ha sido el pasado.

INTERPRETACIÓN DE LAS DIRECCIONES DE MEMORIA

¿Cómo se interpreta una dirección de memoria?

R/

- Es decir, **¿qué objeto es accedido como una función de la dirección y la longitud?**

Todas las computadoras explicadas en este curso están direccionadas por bytes.

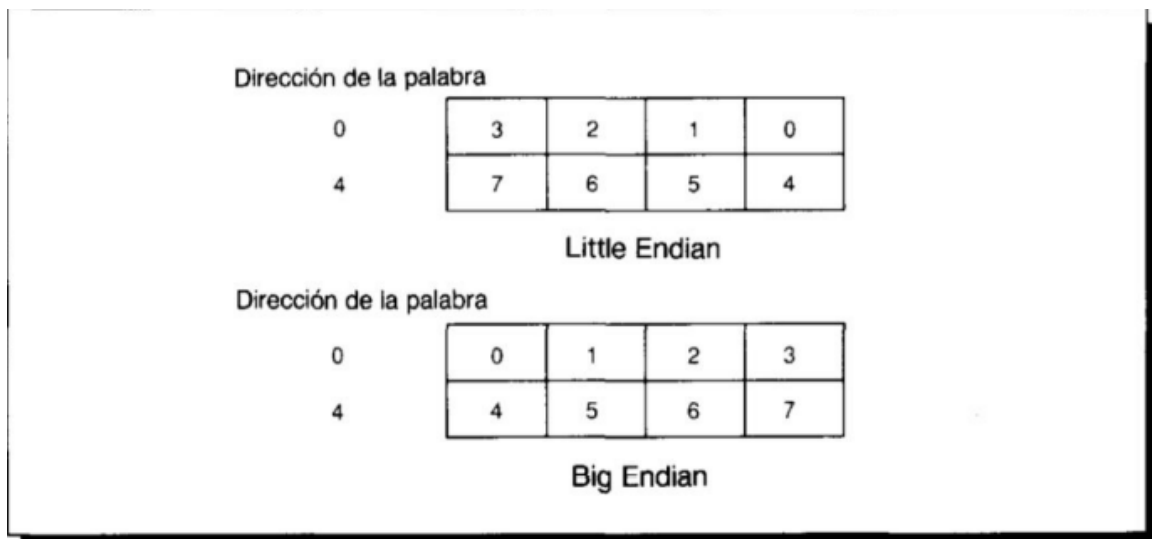
- Proporcionan accesos a bytes (8 bits),
- Medias palabras (16 bits) y
- Palabras (32 bits).
- La mayoría de las máquinas también proporcionan accesos a dobles palabras (64 bits).

EXISTEN DOS CONVENIOS PARA CLASIFICAR LOS BYTES DE UNA PALABRA.

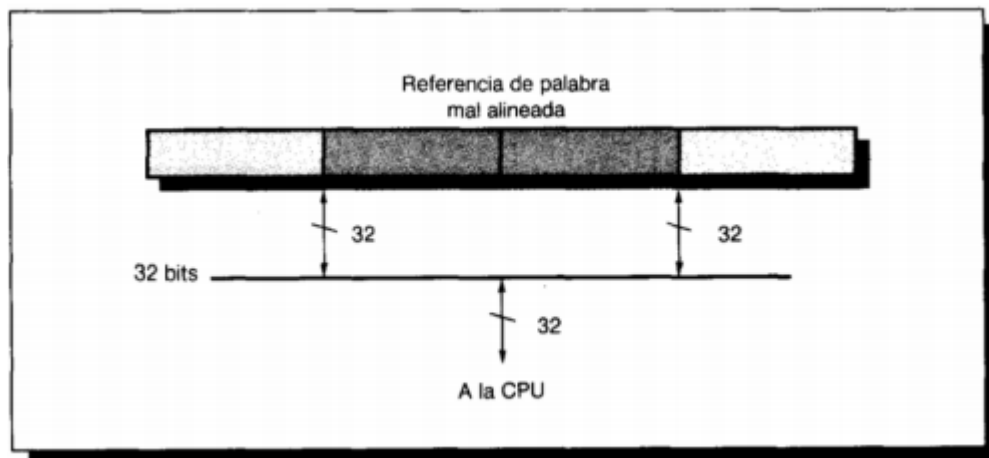
- Hay dos convenios diferentes para clasificar los bytes de una palabra.
- El orden de bytes ((Little Endian)) (Pequeño Endian) coloca el byte cuya dirección es «x ... xOO» en la posición menos significativa de la palabra (little end -extremo pequeño-).
- El orden de bytes «Big Endian» (Gran Endian) coloca el byte cuya dirección es «x ... xOO» en la posición más significativa de la palabra (big end -extremo grande-).

“En el direccionamiento <<Big Endian>>, la dirección de un dato es la dirección del byte más significativo; mientras que en el <<Little Endian>>, es la del byte menos significativo”

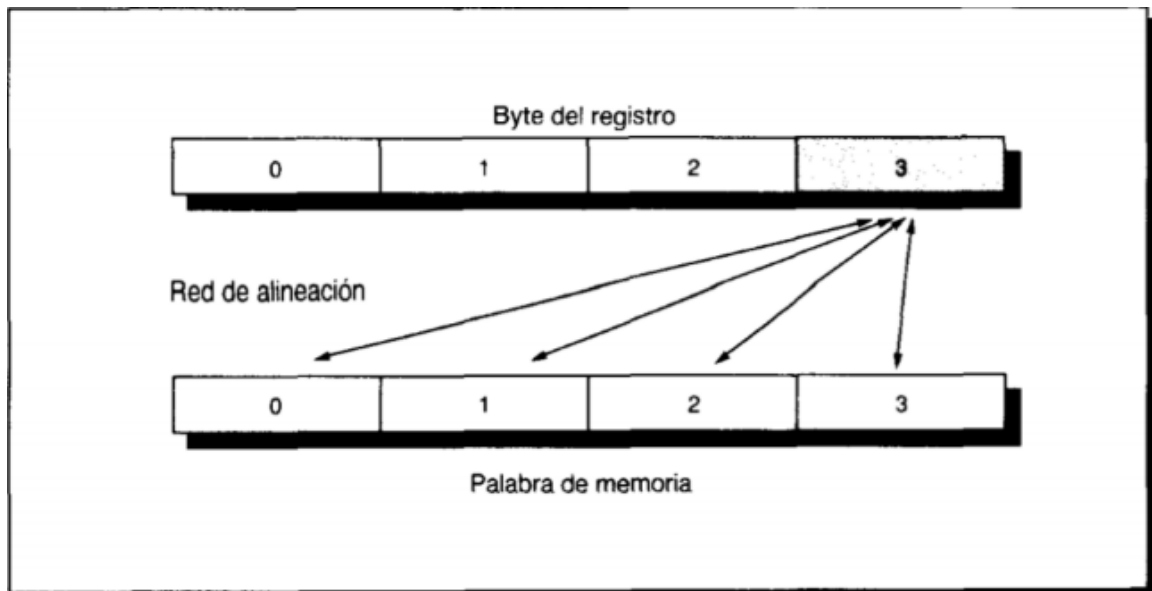
El sistema **big-endian o Big Endian** adoptado por [Motorola](#) entre otros, consiste en representar los bytes en el orden "natural": así el valor [hexadecimal](#) 0x4A3B2C1D se codificaría en memoria en la secuencia {4A, 3B, 2C, 1D}. En el sistema **little-endian o Little Endian** adoptado por [Intel](#), entre otros, el mismo valor se codificaría como {1D, 2C, 3B, 4A}.



- ¿Por qué se diseña una máquina con restricciones de alineación?
- La no alineación causa complicaciones de hardware, ya que la memoria, normalmente, está alineada sobre una frontera de palabras. Un acceso no alineado en memoria, por tanto, tendrá múltiples referencias a una memoria alineada.
- La imagen que se mostrara a continuación muestra qué ocurre cuando se realiza un acceso a una palabra no alineada en un sistema con un bus de 32 bits a memoria:



- Se requieren dos accesos para obtener la palabra. Por tanto, incluso en máquinas que permiten accesos no alineados, los **programas con accesos alineados se ejecutan más rápidamente**.
- En algunas máquinas, un byte o media palabra no afecta a la parte superior de un registro.
- Para los almacenamientos, solamente pueden ser alterados los bytes de memoria afectados. En la imagen que se mostrara a continuación, muestra la red de alineamiento para cargar o almacenar un byte desde una palabra de memoria en un registro. Aunque todas las máquinas permiten accesos a memoria de bytes y medias palabras.



- Los direccionamientos de memoria que utilizaremos para nuestra simulación dependerán del **contador del programa(PC)**, por depender del contador del programa se conocen como **direccionamientos relativos al PC**. Estos modos de direccionamiento se utilizan principalmente para especificar direcciones de código en instrucciones de transferencia de control.