

Arquitectura de un Sistema Informático.

- Los sistemas informáticos actuales, ya sean computadores personales, grandes supercomputadores o smartphones, tienen como base las arquitecturas de Von Neumann y Harvard.
- El modelo de Von Neumann consta de las siguientes partes:
 - Unidad de procesamiento(CPU): se encarga de la ejecución e interpretación de **instrucciones** y datos formada por unidad aritmético lógica (ALU), unidad de control y **registros** de almacenamiento.
 - Memoria: almacena instrucciones y datos.
 - Dispositivos de entrada/salida: elementos que actúan de interfaz con el resto de partes.

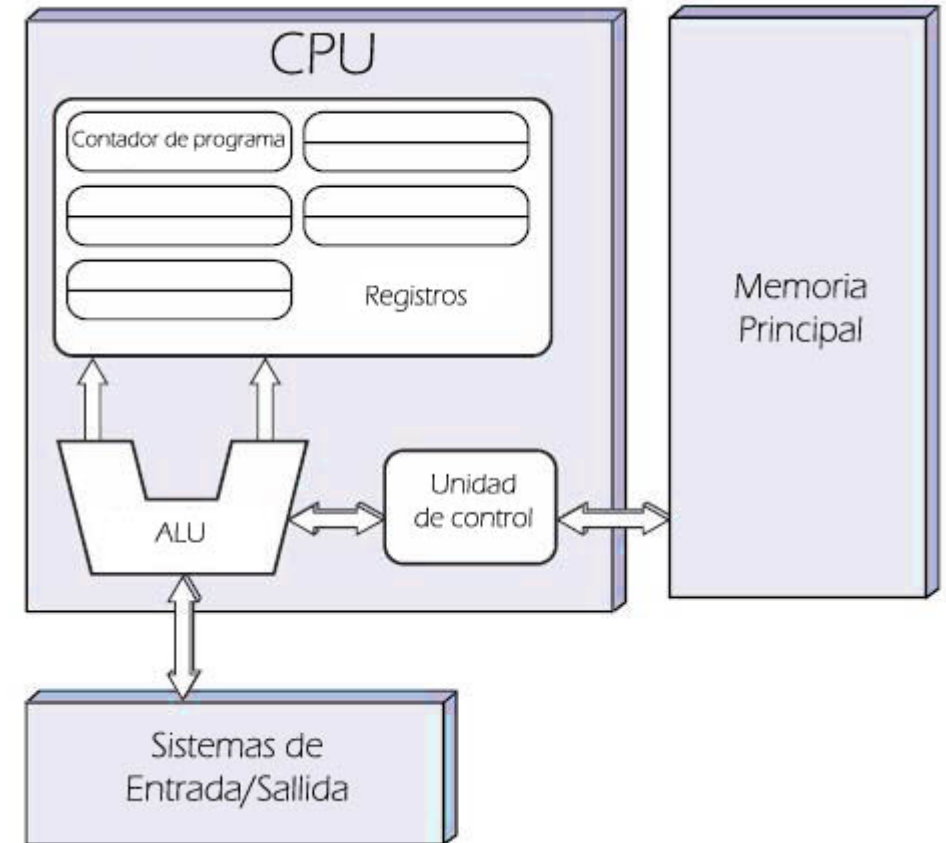
Arquitectura von Neumann. I

En el modelo de von Neumann las diferentes unidades funcionales se interconectan mediante **buses**.

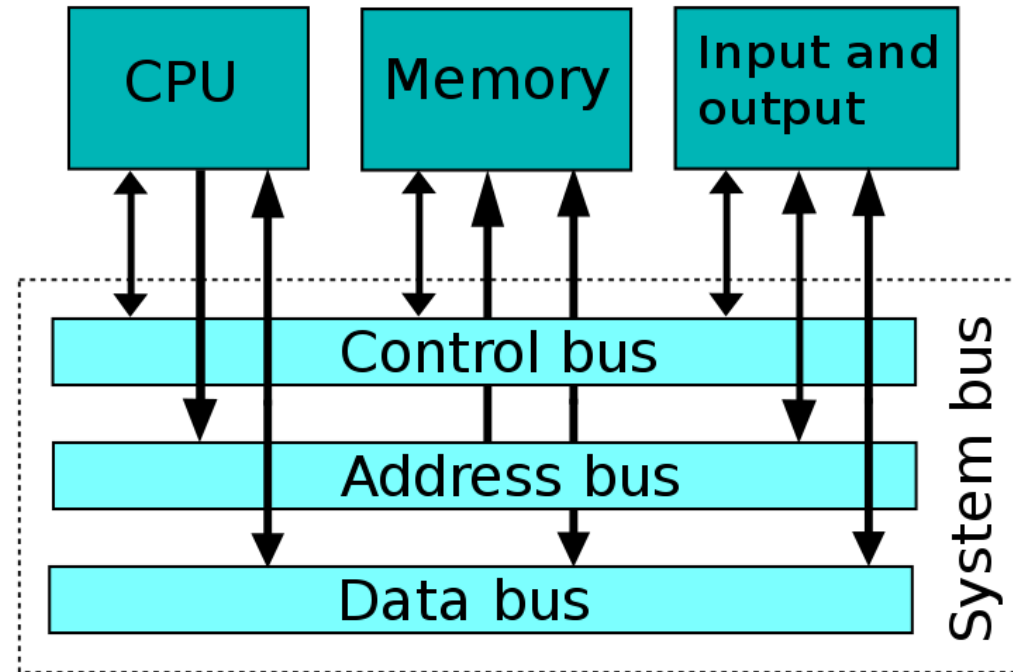
Buses de instrucciones: líneas de comunicación que transmiten instrucciones.

Buses de **datos**: líneas de comunicación que transmiten únicamente datos.

Buses de **direcciones**: líneas de comunicación empleadas para acceder a las distintas memorias, indicando una dirección de acceso de lectura o escritura.



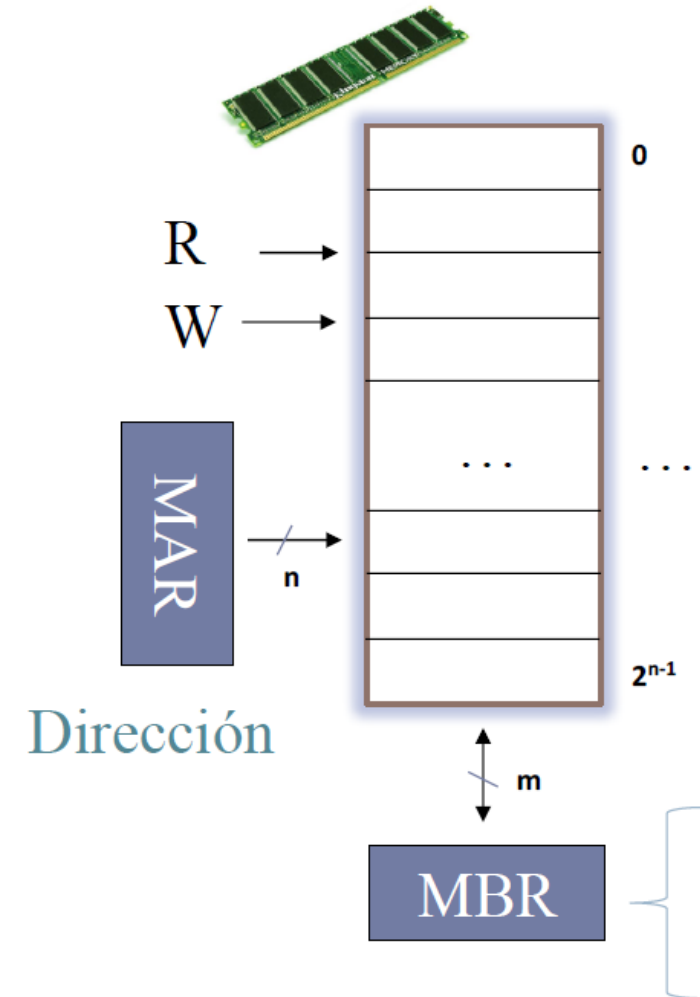
Arquitectura von Neumann. II



- <http://vnsimulator.altervista.org/>
- <https://asir-fuw.github.io/Makinito/>

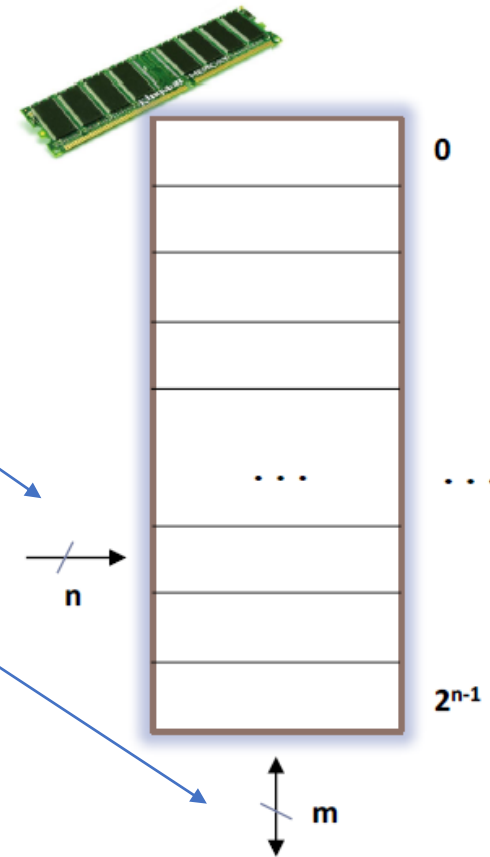
Memoria principal.

- Registro de direcciones (MAR, Memory Address Register)
- Registro de datos (MBR, Memory Buffer Register)
- Señales de control
 - R- Lectura (Read)
 - W- Escritura (Write)



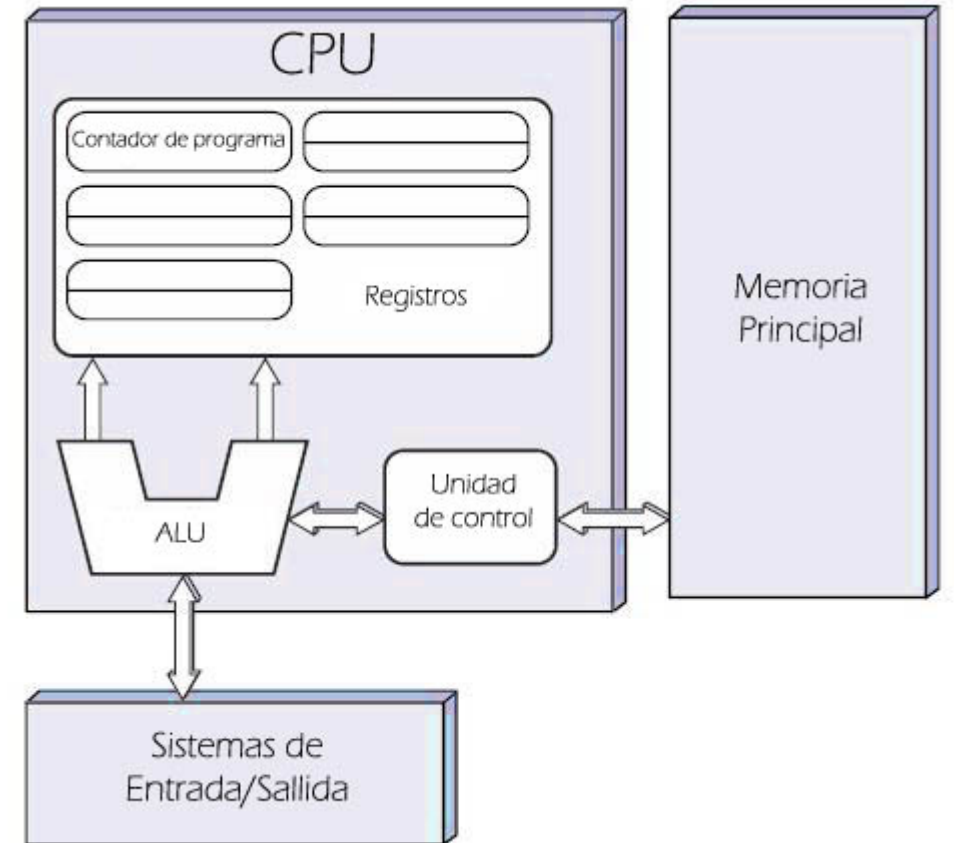
Espacio de direcciones vs. tamaño de palabra

- Espacio de direcciones:
 - Número de posiciones = 2^n
- Tamaño de cada posición:
 - Número de bits por posición = m
- Ejercicio.
 - $n = 14$
 - $m = 16$
 - Calcular la capacidad de la memoria.



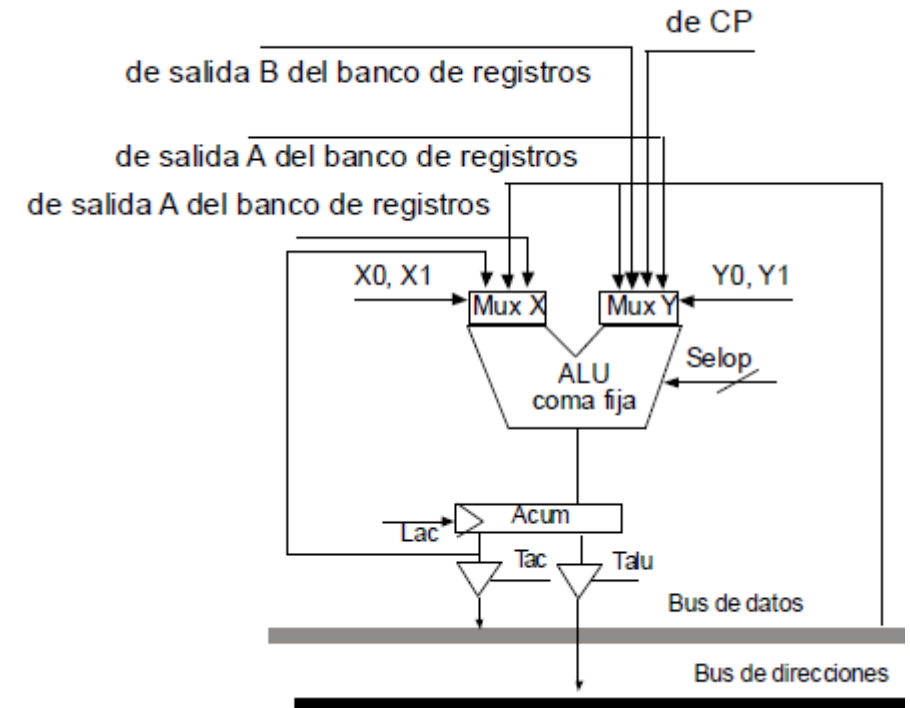
CPU. I

- Registros:
 - Almacenan datos binarios, acceso rápido.
 - De tamaño fijo.
 - De **propósito general** (programas) o **específicos** (acumulador, contador de programa, puntero a memoria, etc.)
- DataPath
 - Red interna que comunica la UC con las otras unidades y registros.
 - Mueve datos entre los diferentes componentes.
 - Controlada por un reloj.



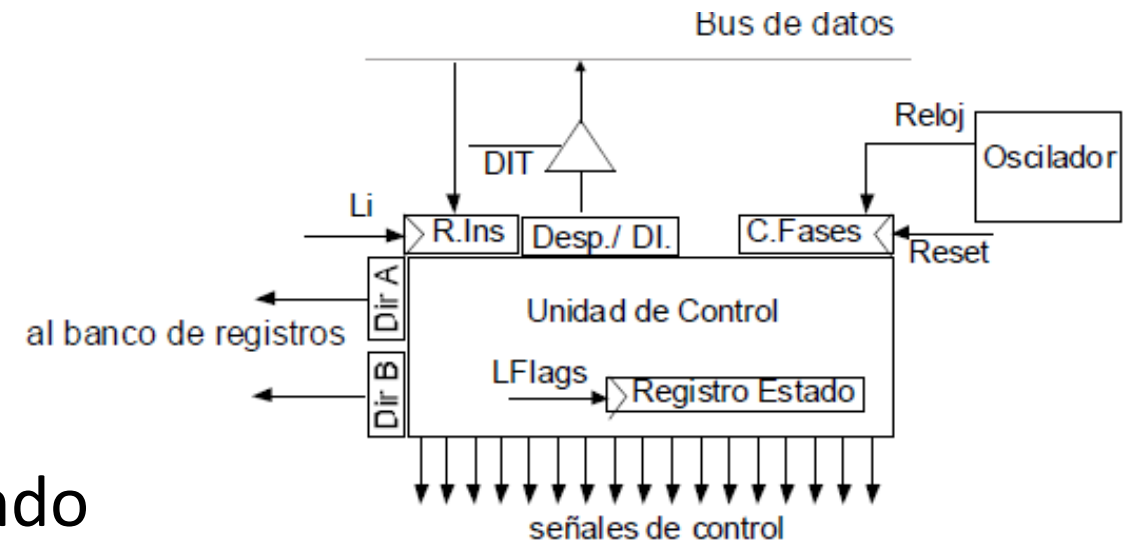
CPU. II

- La unidad aritmético-lógica consta de los siguientes elementos:
 - Un operador con señales de control para selección de la operación a realizar.
 - Multiplexores de varias entradas a 1 salida, que seleccionan entre los posibles operandos
 - Un **registro acumulador** que permite almacenar resultados intermedios.
 - La salida del registro acumulador, puede transferirse al bus de datos o al bus de direcciones, según indiquen las señales de control.
- ¿Operaciones típicas de una ALU? ¿Qué es una FPU?



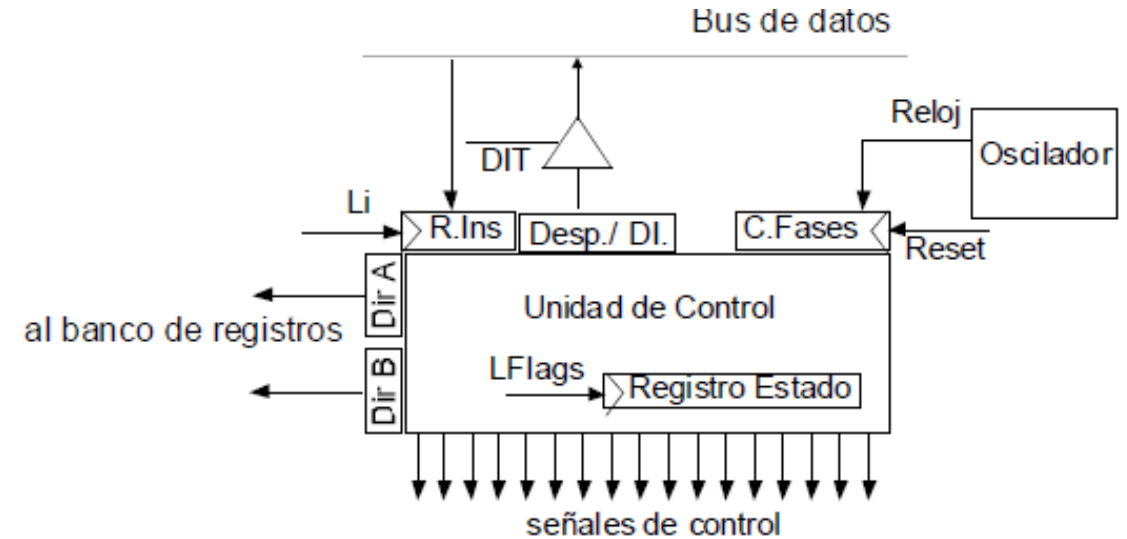
CPU. III

- La UC necesita del código de operación de la instrucción, el estado del sistema y las señales de E/S, al ritmo del contador de fases.
- La UC es un circuito que genera las señales de control necesarias para ejecutar una instrucción.
- El diseño de la UC se realiza definiendo todas las señales de control en función de todas las posibles entradas.



CPU. IV

- La unidad de control necesita una serie de registros:
 - **Registro de Instrucciones (IR)** guarda la instrucción actual.
 - **Contador de programa (PC)** contiene la dirección de la siguiente instrucción a ejecutar.
 - **Registro de estado**, que contiene los flags. Este registro se carga selectivamente con varias señales de control activas por flanco y provenientes de la ALU.
 - **Registro contador de fases**, gobernado por el reloj del sistema y que se puede poner a 0 con la señal RESET.

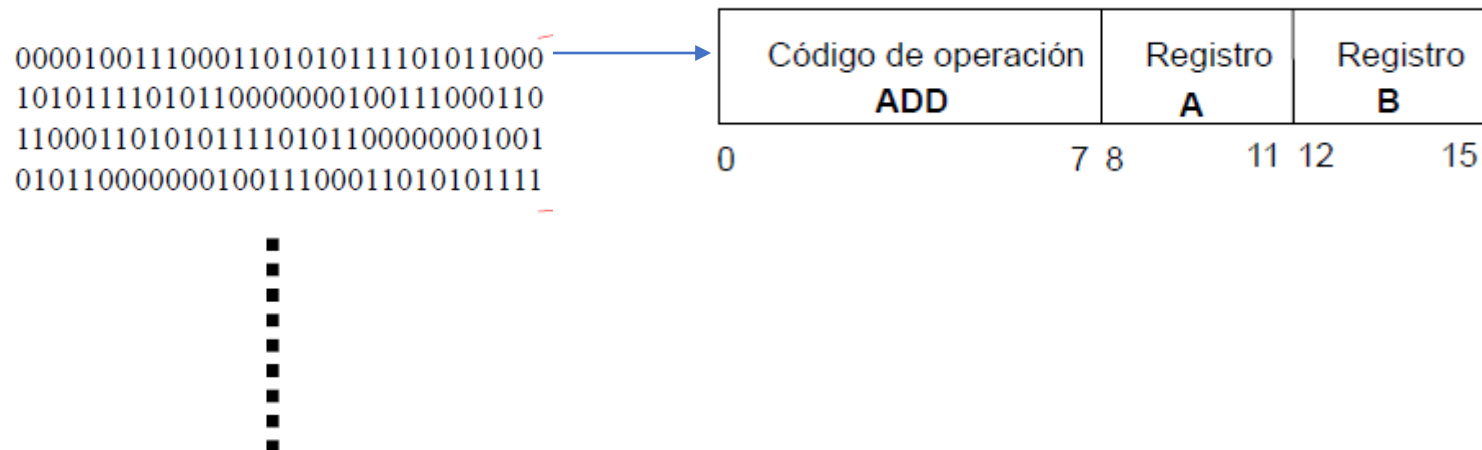


CPU. V

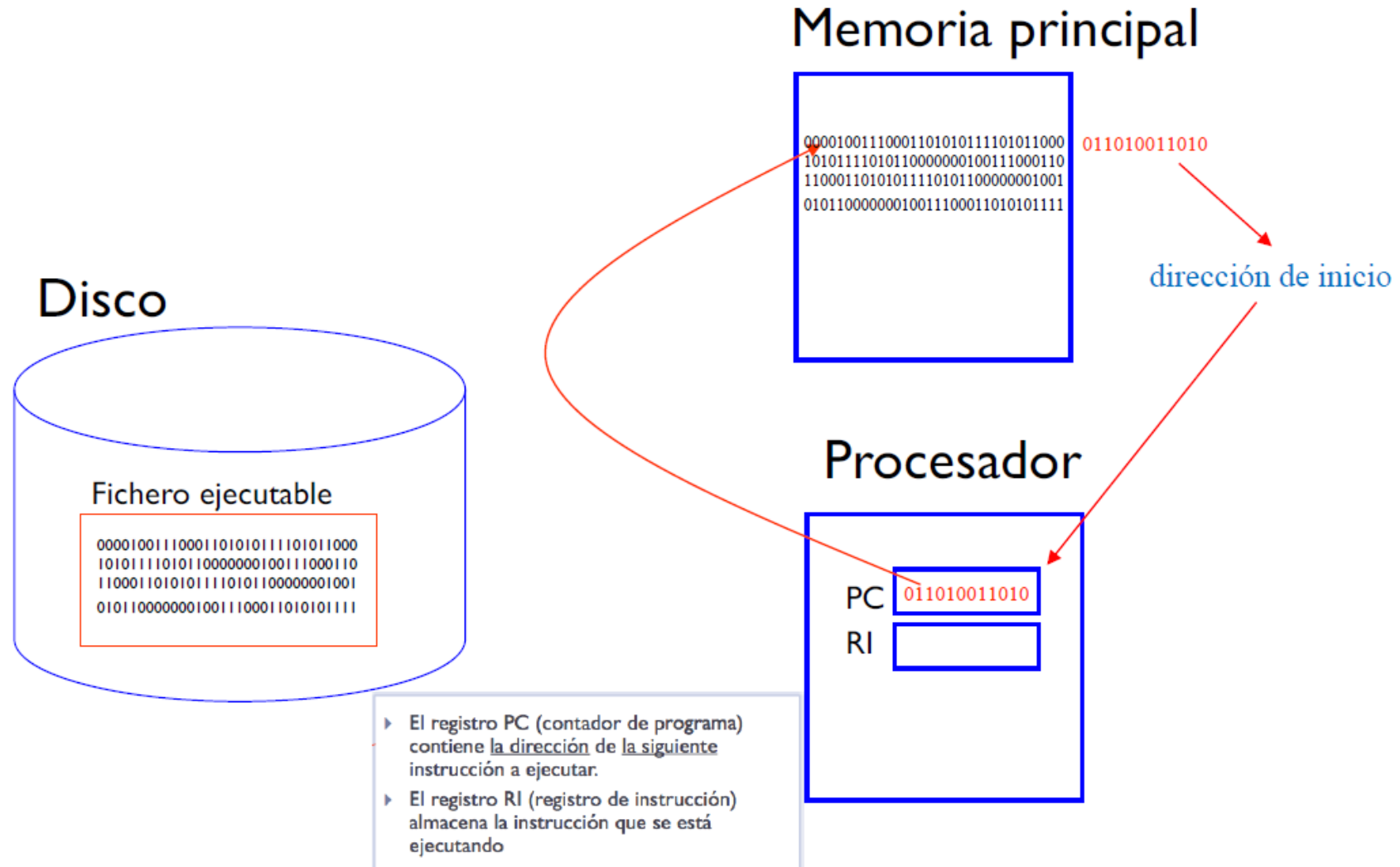
- El comportamiento de un computador es síncrono. Está gobernado por un reloj
- La ejecución de una instrucción se divide en:
 1. Fase de búsqueda de la instrucción o fase de fetch
 2. Decodificar la instrucción
 3. Ejecución de la instrucción y en caso necesario:
 - 3.1 Leer los operandos
 - 3.2 Realizar la operación
 - 3.3 Almacenar el resultado
 - 3.4 Actualizar el registro de estado
 4. Habrá que actualizar el CP para que apunte a la instrucción siguiente a ejecutar
- Cada una de las fases se realiza en un número determinado de periodos de reloj

Programa.

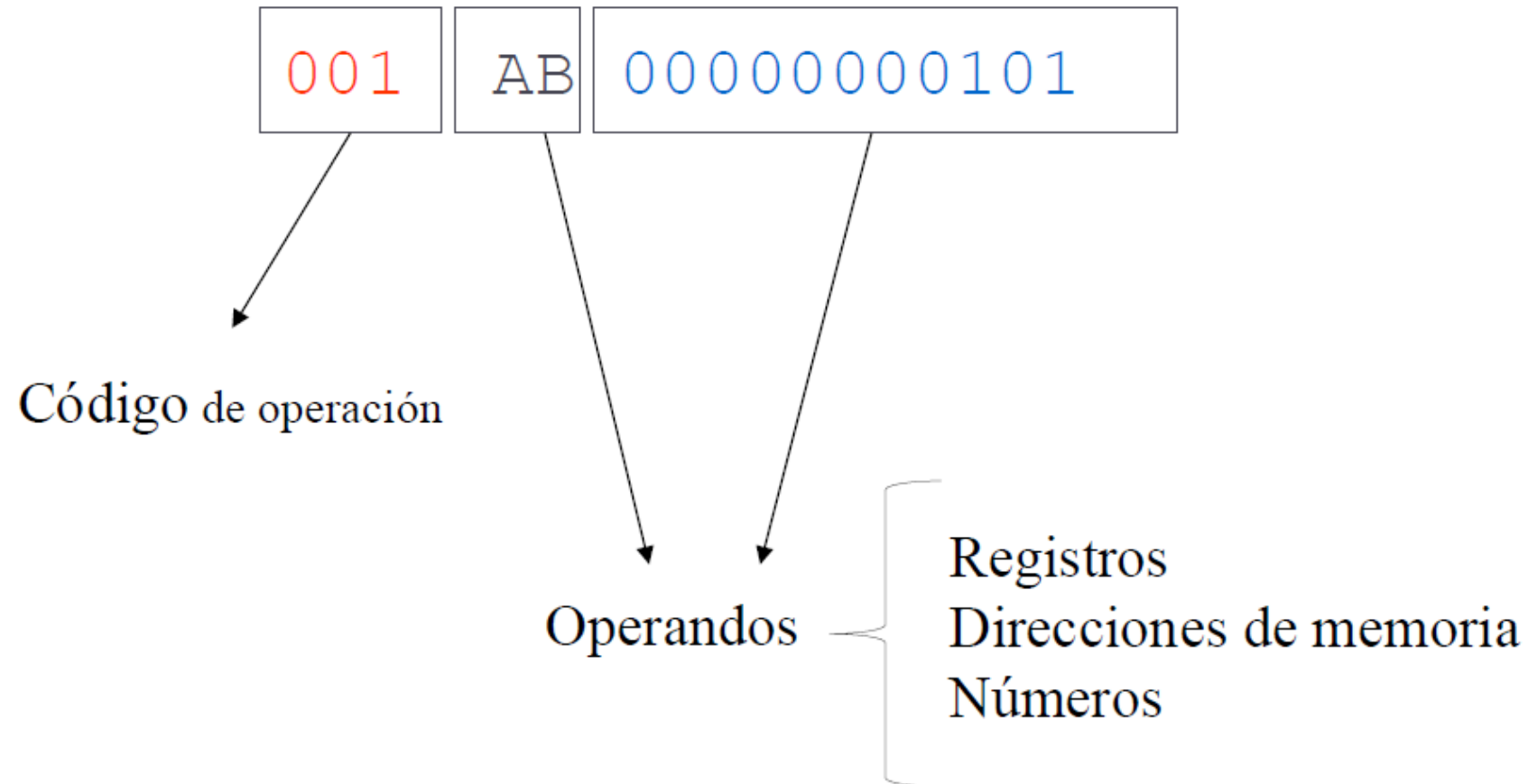
- Secuencia consecutiva de instrucciones máquina.
- Instrucción máquina: operación elemental que puede entender y ejecutar directamente unidad de control.
 - Codificación en binario y almacenado en la memoria.



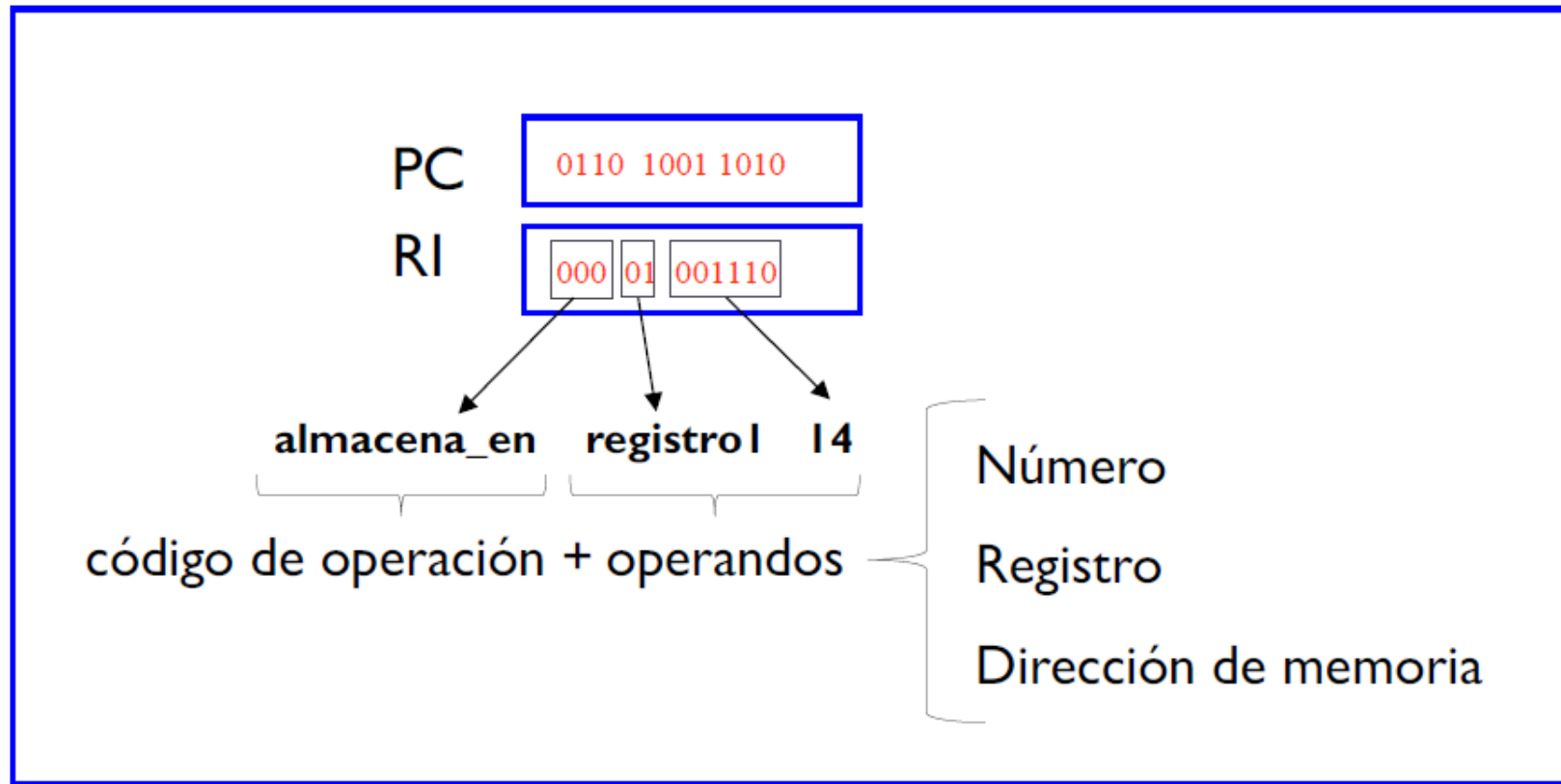
Ejecución de instrucciones.



Formato de una instrucción.



Instrucción en la CPU.



Ejemplo de juego de instrucciones. I

- Ejemplo de conjunto de instrucciones con las siguientes características:
 - Tamaño de una posición de memoria: 16 bits
 - Tamaño de la instrucción: 16 bits
 - Código de operación: 3 bits
 - ¿Cuántas instrucciones diferentes puede tener este computador?
- Número de registros de propósito general: 4
- Identificadores simbólicos:
 - R0
 - R1
 - R2
 - R3
- ¿Cuántos bits se necesitan para representar estos 4 registros?

Ejemplo de juego de instrucciones. II

Instrucción	Descripción
000CCAABBXXXXXXXX	Suma el registro AA con el BB y deja el resultado en CC
001AA00000000101	Almacena en el registro AA el valor 00000000101
010AA00000001001	Almacena en el registro AA el valor almacenado en la posición de memoria 00000001001
011AA00000001001	Almacena en la posición de memoria 00000001001 el contenido del registro AA
1000000000001001	Se salta a ejecutar la instrucción almacenada en la posición de memoria 0000000001001
101AABB000001001	Si el contenido del registro AA es igual al del registro BB se salta a ejecutar la instrucción almacenada en 000001001

Siendo A,B,C,D,E,F = 0 ó 1

Ejemplo de programa cargado en memoria.

Memoria principal

Dirección	Contenido
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

Generación y carga de un programa.

```
i=0;  
s = 0;  
while (i < 4)  
{  
    s = s + 1;  
    i = i + 1;  
}
```

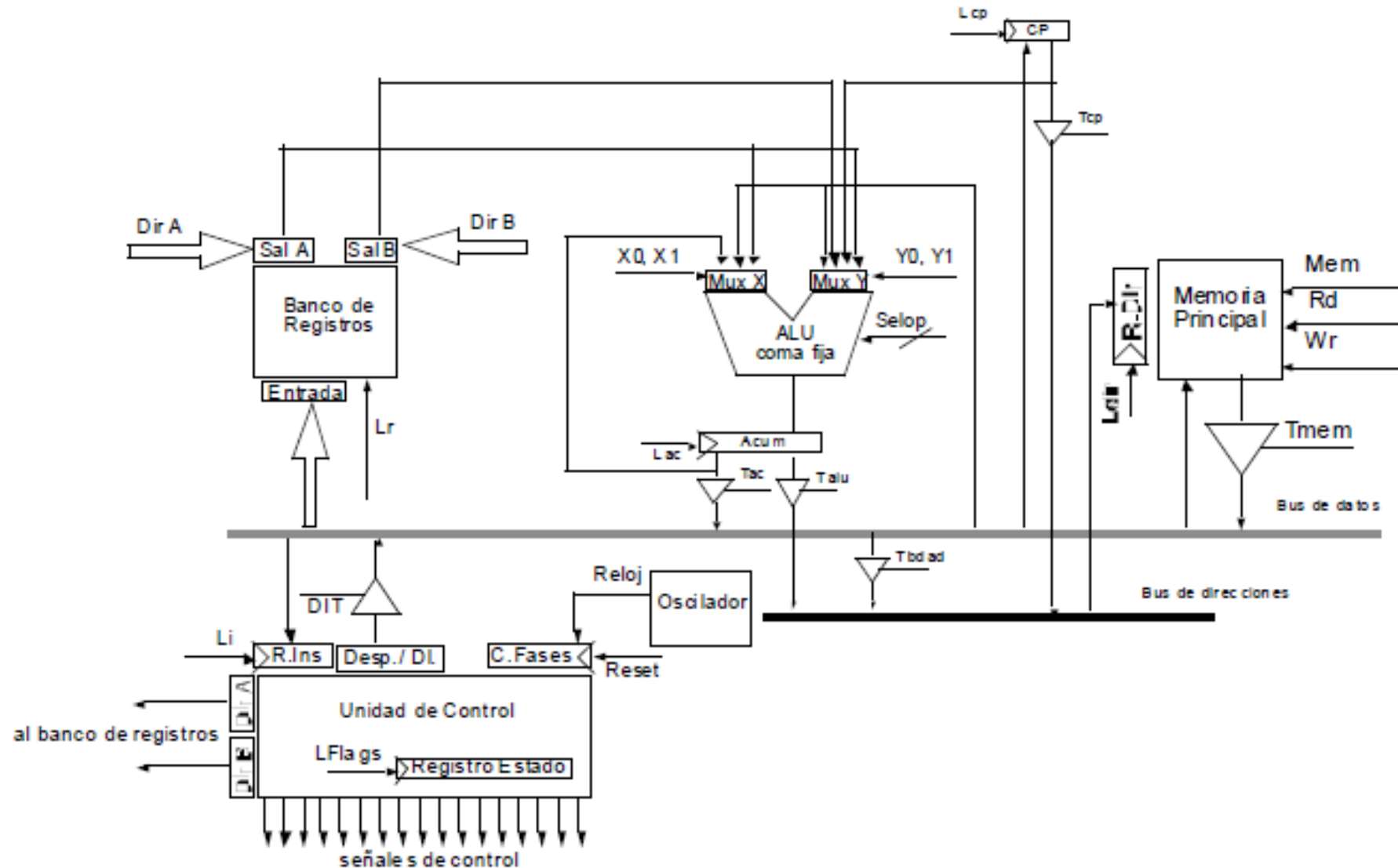


```
li R0, 0  
li R1, 4  
li R2, 1  
li R3, 0  
lazo: beq R0, R1, fin  
      add R3, R3, R2  
      add R0, R0, R2  
      b lazo  
fin:  sw R3, 100000
```

000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000



Esquema máquina von Neumann.



Ejercicio.

- Considere un hipotético computador con un ancho de palabra de 20 bits con 60 registros. Responda a las siguientes preguntas:
 - ¿Cuántos bits se emplean para las direcciones de memoria?
 - ¿Cuál es el tamaño de los registros?
 - ¿Cuántos bits se almacenan en cada posición de memoria?
 - ¿Cuántas posiciones de memoria se pueden direccionar? Expresa el resultado en KB. e)
¿Cuántos bits se necesitan para identificar a los registros?

Arquitectura Harvard.

- Trabajo de clase: Arquitectura Harvard.
 - Estructura y funcionamiento.
 - Diferencias con la arquitectura von Neumann.
 - Ventajas e inconvenientes.
 - Arquitectura de las máquinas actuales.
 - Fuentes documentales.
- Entrega en Classroom.
- Fecha límite 7-10-21.