



DEPARTAMENTO DE INFORMÁTICA TRELEW

Facultad de Ingeniería - Universidad Nacional de la Patagonia San Juan Bosco

 **<http://www.dit.ing.unp.edu.ar>**

 **DIT-Dpto.Informática Trelew**

ELEMENTOS DE INFORMÁTICA

Departamento de Informática Trelew - Facultad de Ingeniería
Universidad Nacional de la Patagonia San Juan Bosco



- Computadora Electrónica Digital

<http://es.wikipedia.org/wiki/Computadora#Funcionamiento>

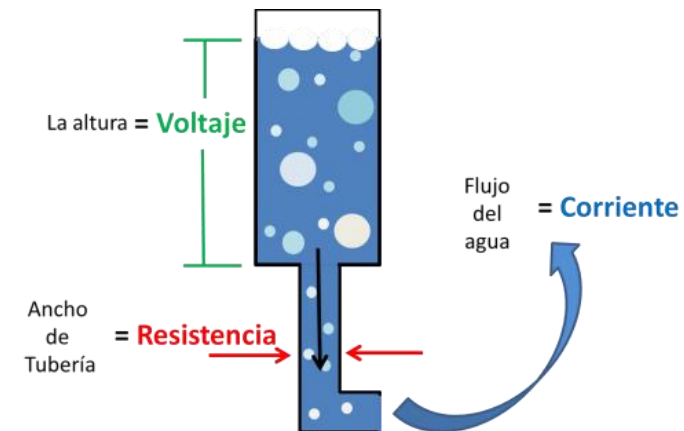
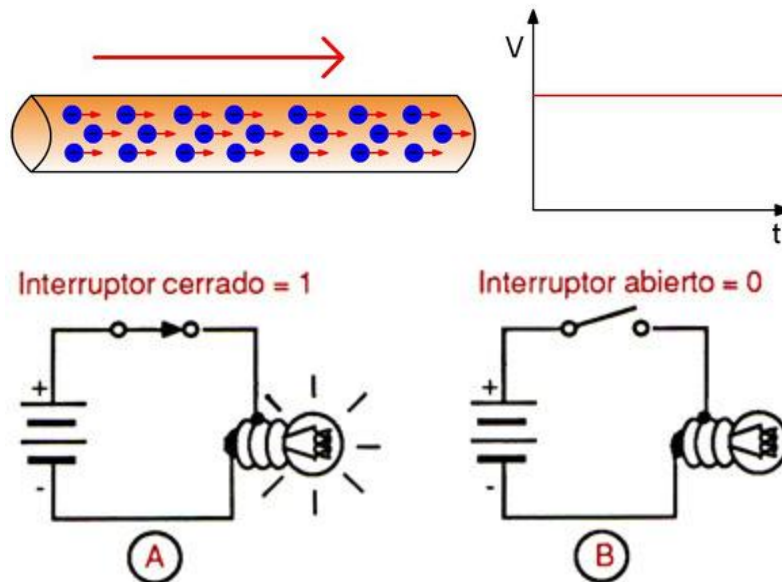
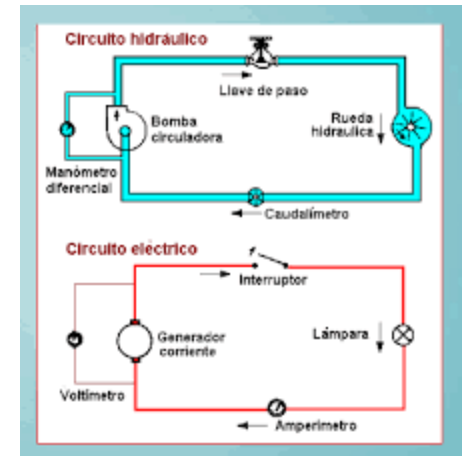
Anexo 1: Capítulo 5 al 8

Corriente Eléctrica

En Informática se utiliza un soporte (Hardware) que funciona con circuitos electrónicos, donde los transistores, capacitores y las resistencias son los componentes más habituales. Estos últimos dispositivos están diseñados para que circule una corriente eléctrica (I) por ellos, provocando un potencial eléctrico (V) en puntos que son de interés.

Corriente eléctrica.

Es el flujo de carga eléctrica que recorre un material. Se debe al movimiento de las cargas (normalmente electrones) en el interior del mismo. Al caudal de corriente (cantidad de carga por unidad de tiempo) se le denomina intensidad de corriente eléctrica (representada comúnmente con la letra I).



Fenómenos analógicos y digitales

Fenómenos analógicos.

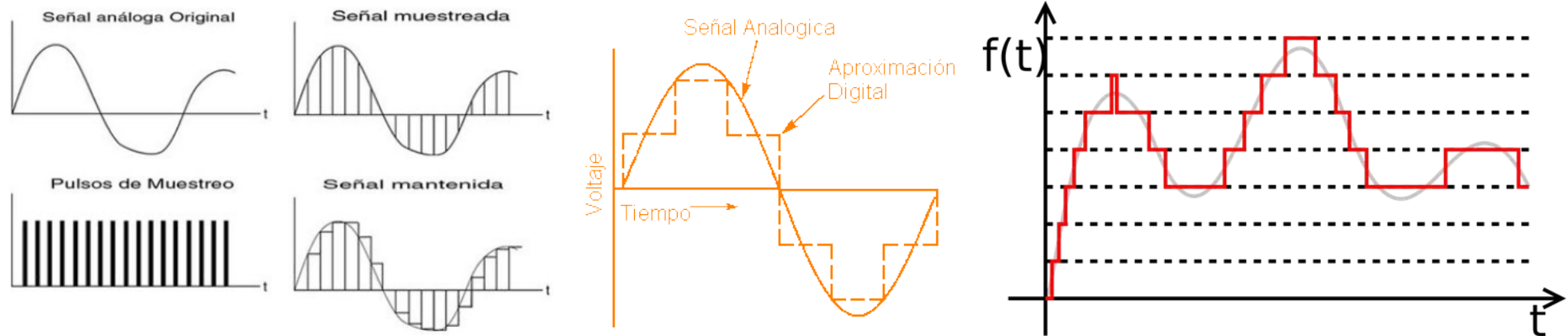
En la naturaleza las variaciones, sean cuales fueren, se producen en forma continua, es decir no presentan grandes diferencias sin haber pasado por valores intermedios muy próximos. Ejemplos: la temperatura, la velocidad del viento, etc. Estos tipos de fenómenos presentan infinitos puntos intermedios entre los valores posibles.

Fenómenos digitales.

Creados por el humano, tienen la característica de tomar sus valores extremos, sin necesidad de pasar por los puntos intermedios.

Las computadoras trabajan en forma digital, utilizando para sus señales internas la ausencia o presencia de electricidad; esto es así debido a que una interferencia externa, por ejemplo un campo electromagnético intenso, podría afectar la tensión de un cableado.

Combinando entonces, las señales digitales, encendido y apagado, pueden representarse letras, números y símbolos. El estado de "encendido" y "apagado" se representa mediante un BIT (abreviatura de binary digit).



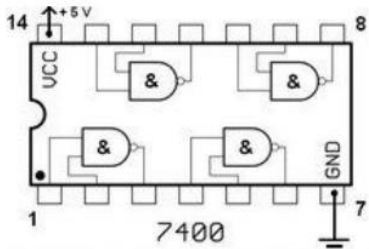
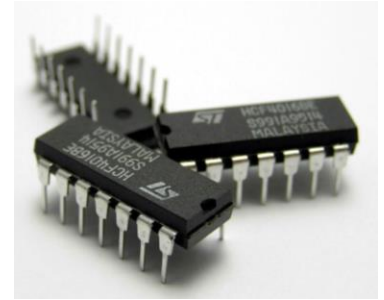
Circuitos Combinacionales

Compuertas electrónicas

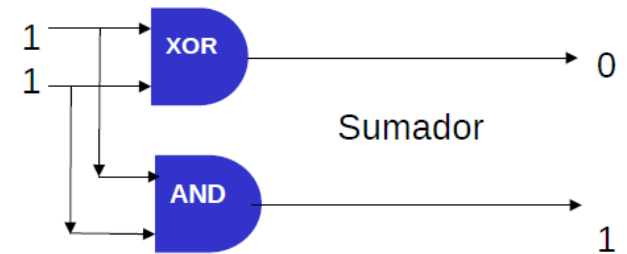
Responden a los valores lógicos en las entradas, la salida está determinada exclusivamente por los valores de las entradas en ese instante.

Si cambia la entrada, cambia la salida.

Los valores pasados de las entradas no influyen en los valores de las salidas.



$$\begin{array}{r} (1) \\ 0 \ 1 \\ + \ 1 \ 1 \\ \hline 1 \ 0 \ 0 \end{array}$$



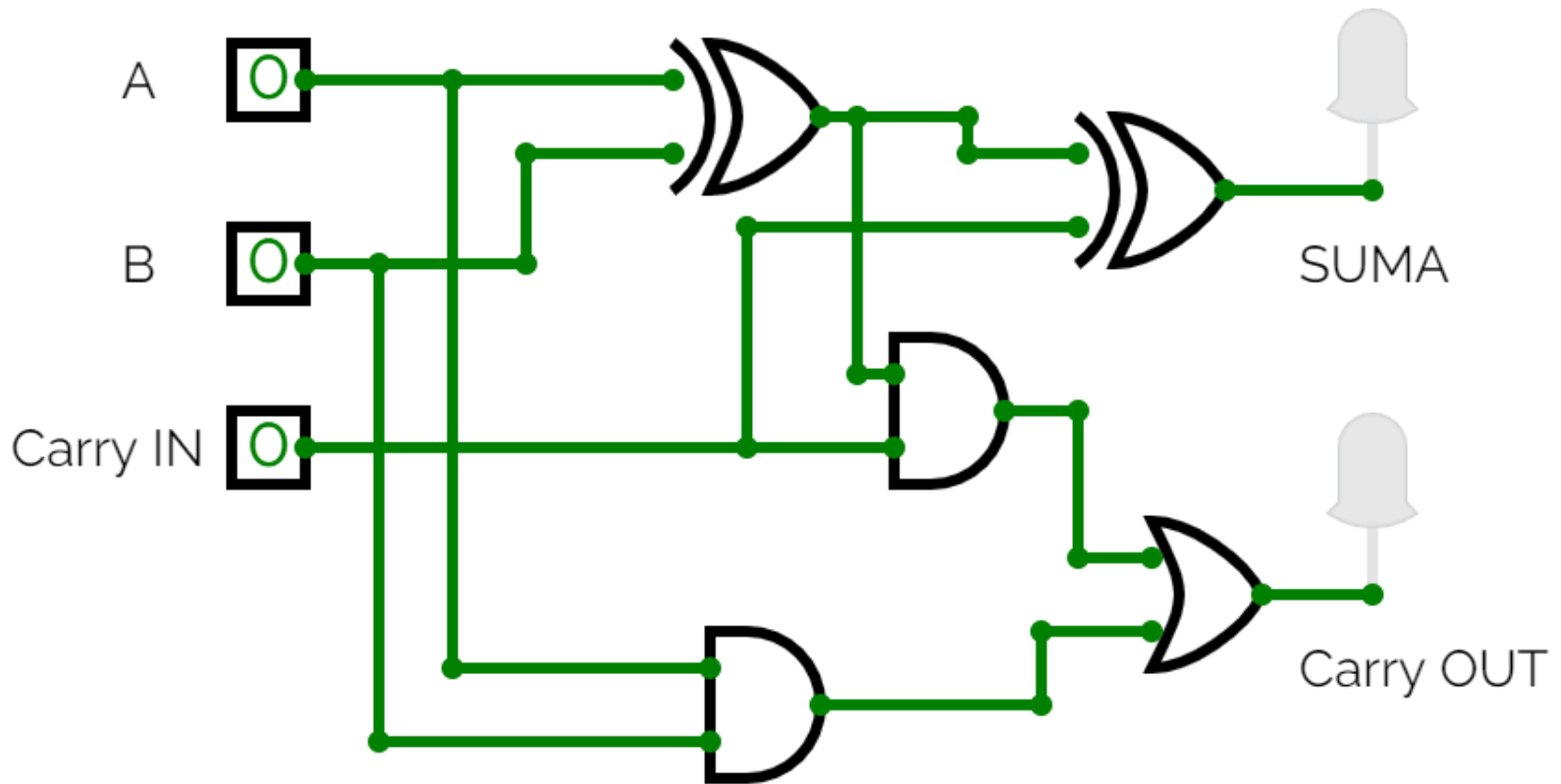
X	Y	AND
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	OR
0	0	0
0	1	1
1	0	1
1	1	1

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

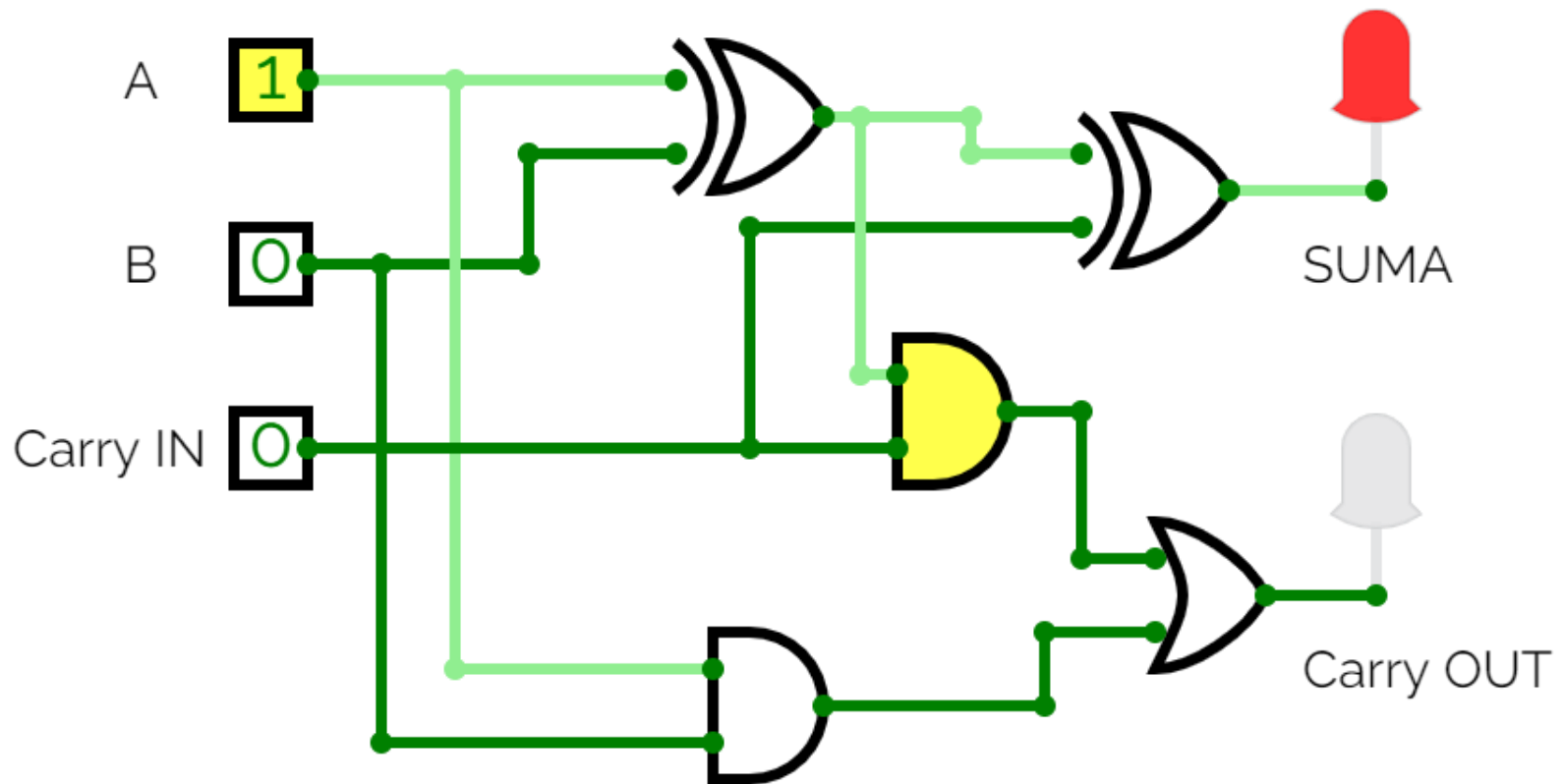
Circuitos Combinacionales

Sumador completo



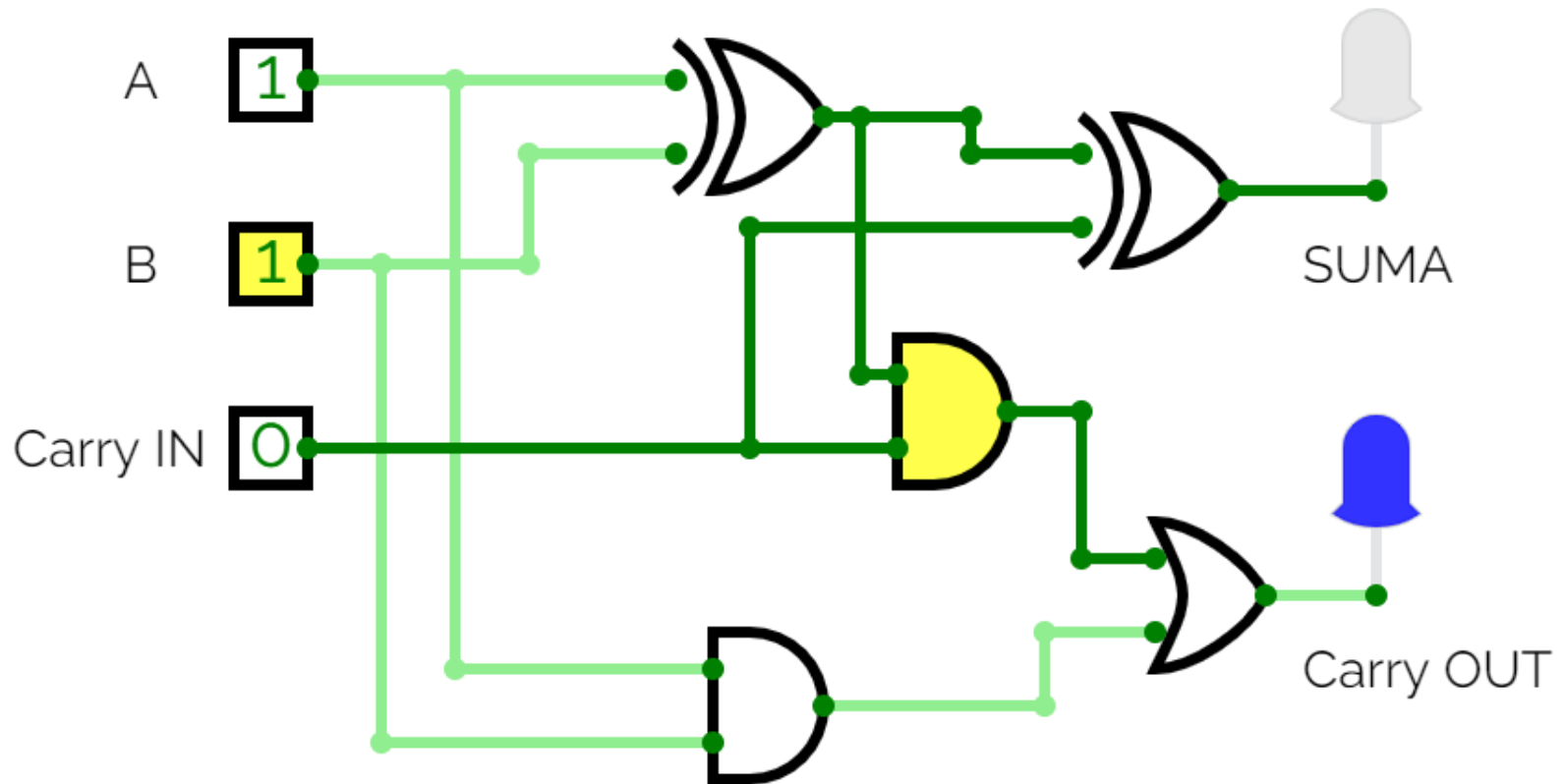
Circuitos Combinacionales

Sumador completo



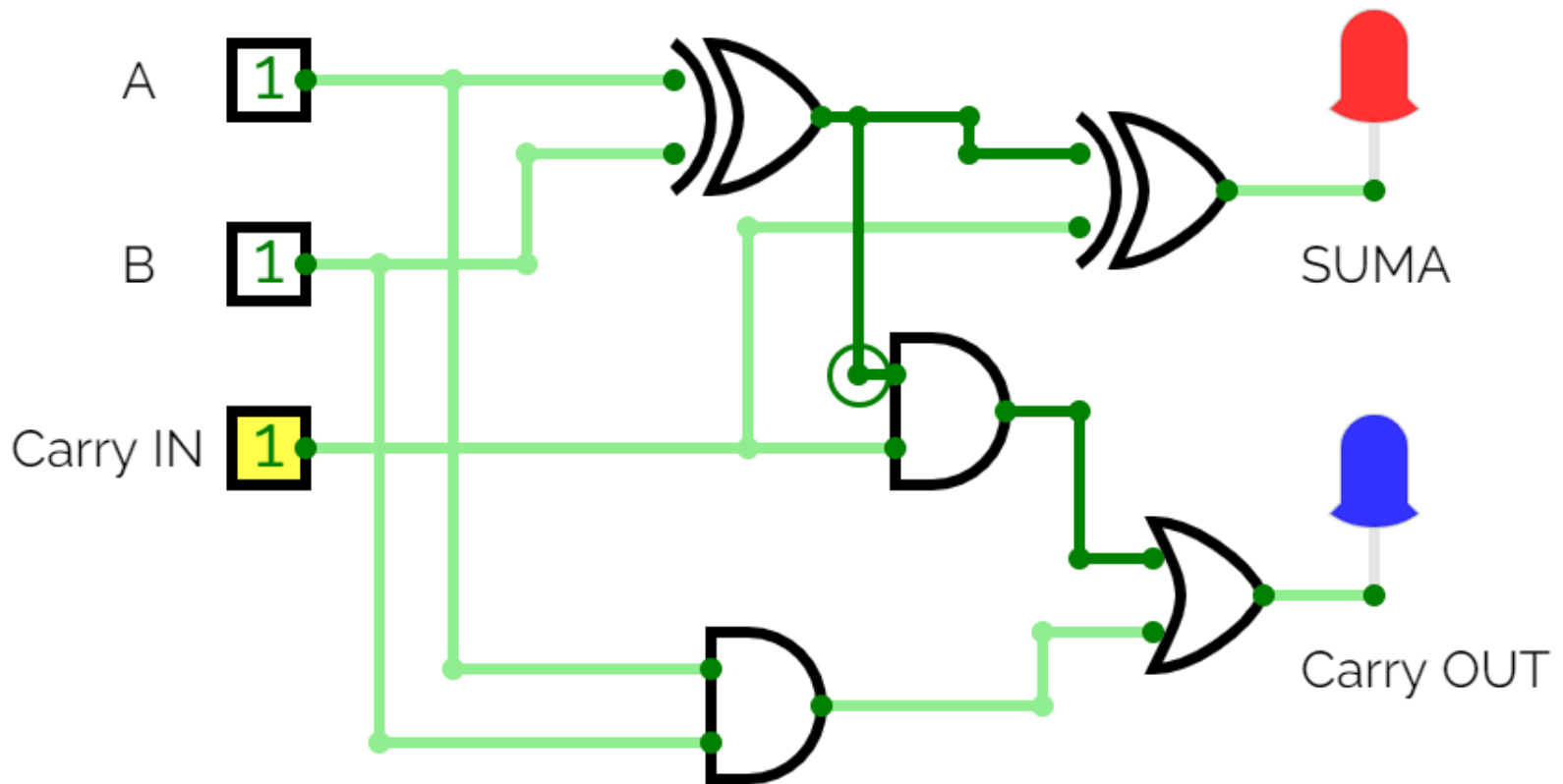
Circuitos Combinacionales

Sumador completo



Circuitos Combinacionales

Sumador completo



Conceptos básicos

Computadora.

- Máquina. Digital. Sincrónica Cálculo numérico. Cálculo lógico. Controlada por programa. Comunicación con el mundo exterior.

Arquitectura.

- Aquellos atributos visibles al programador.
 - Conjunto de instrucciones, números de bits usados para representación de datos, mecanismos de E/S, técnicas de direccionamiento. Ej. ¿Existe la instrucción de multiplicación?
 - Toda la familia Intel x86 comparte la misma arquitectura básica. Lo mismo en la familia IBM 370.

Organización.

- Es como son implementados.
 - Señales de control, interfaces, tecnología de memoria. Ej. ¿Existe una unidad de multiplicación o se realiza por sumas sucesivas?

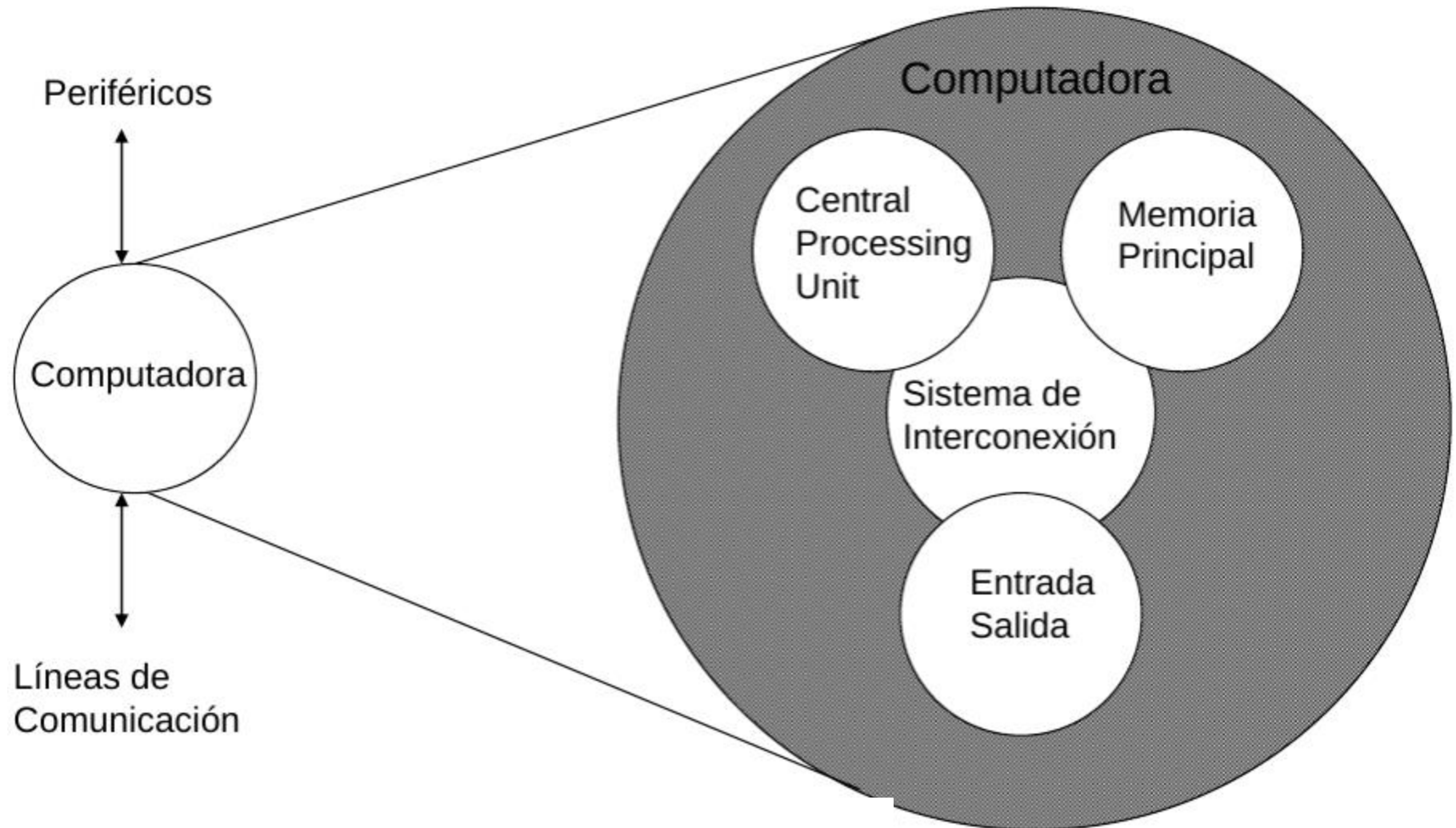
Estructura.

- Es el modo en el cual los componentes se relacionan entre sí.

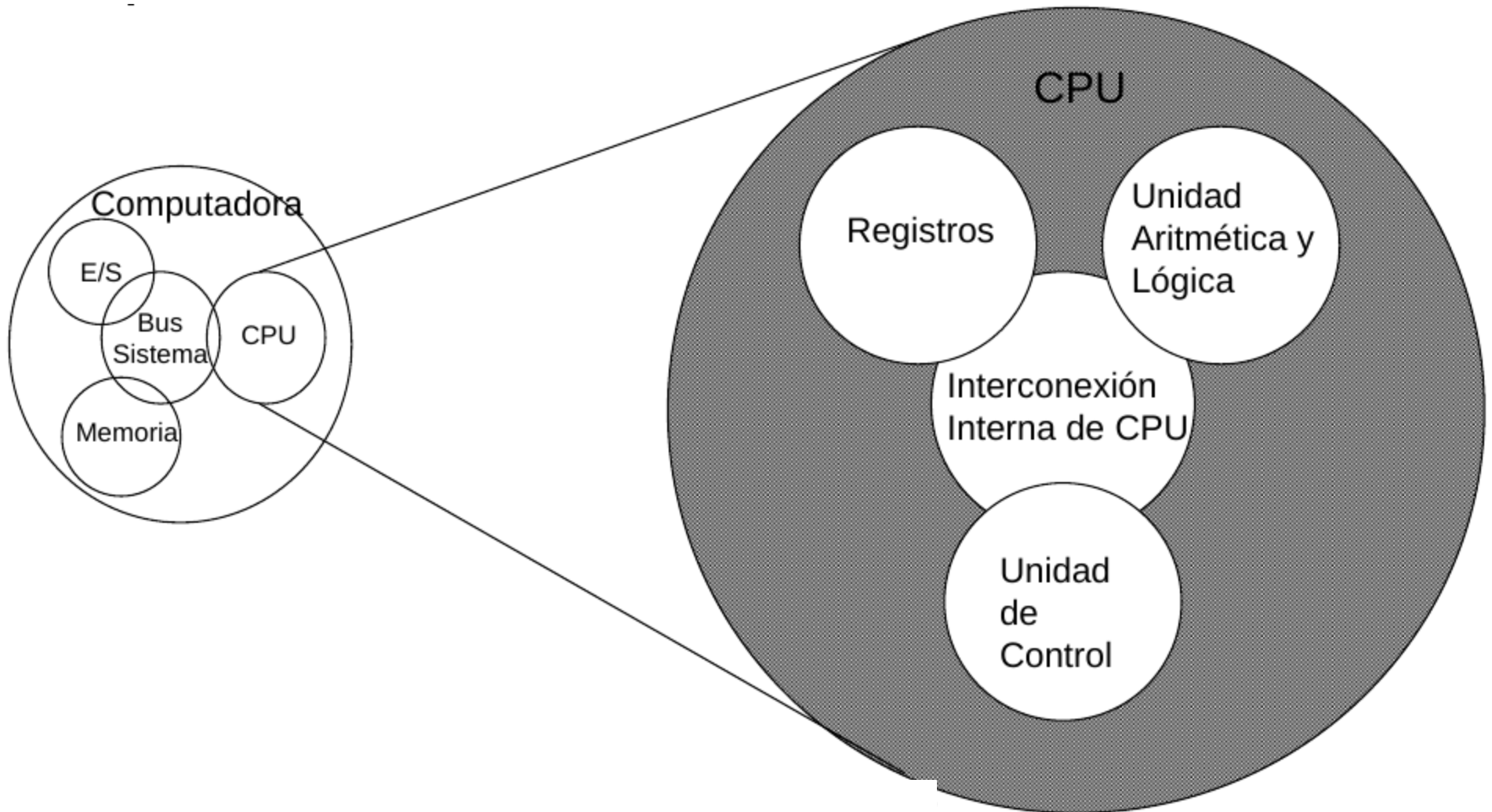
Función.

- Es la operación de los componentes individuales como parte de la estructura.
 - Procesamiento, Almacenamiento, Movimiento y Control de datos.

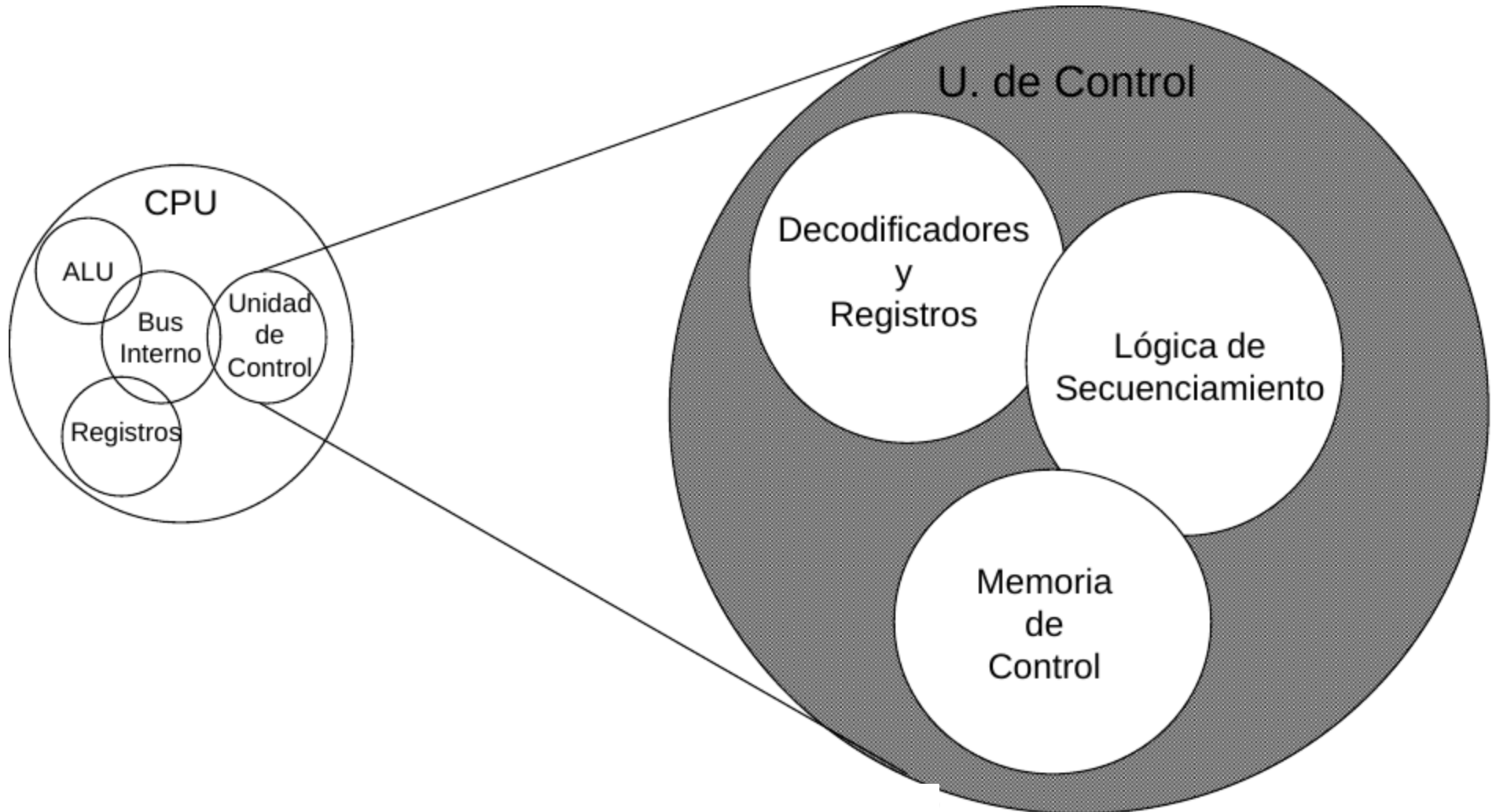
Estructura – Nivel superior



Estructura – CPU

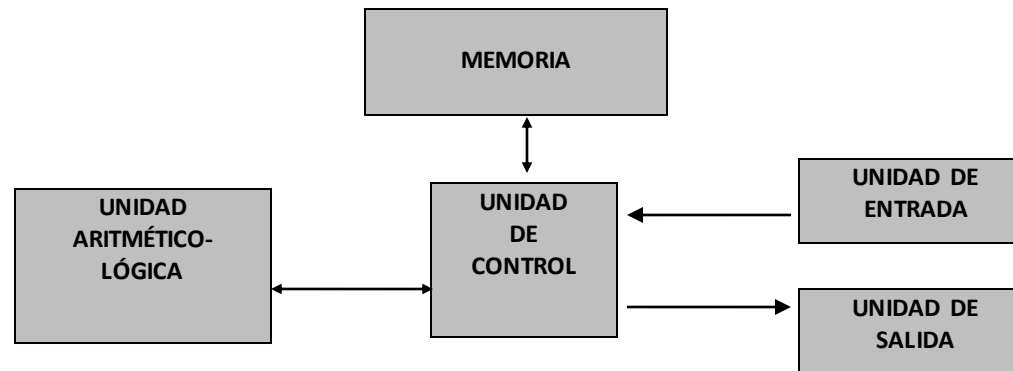


Estructura – Unidad de Control



Estructura – Modelo Von Neumann

MAQUINA DE VON NEUMANN

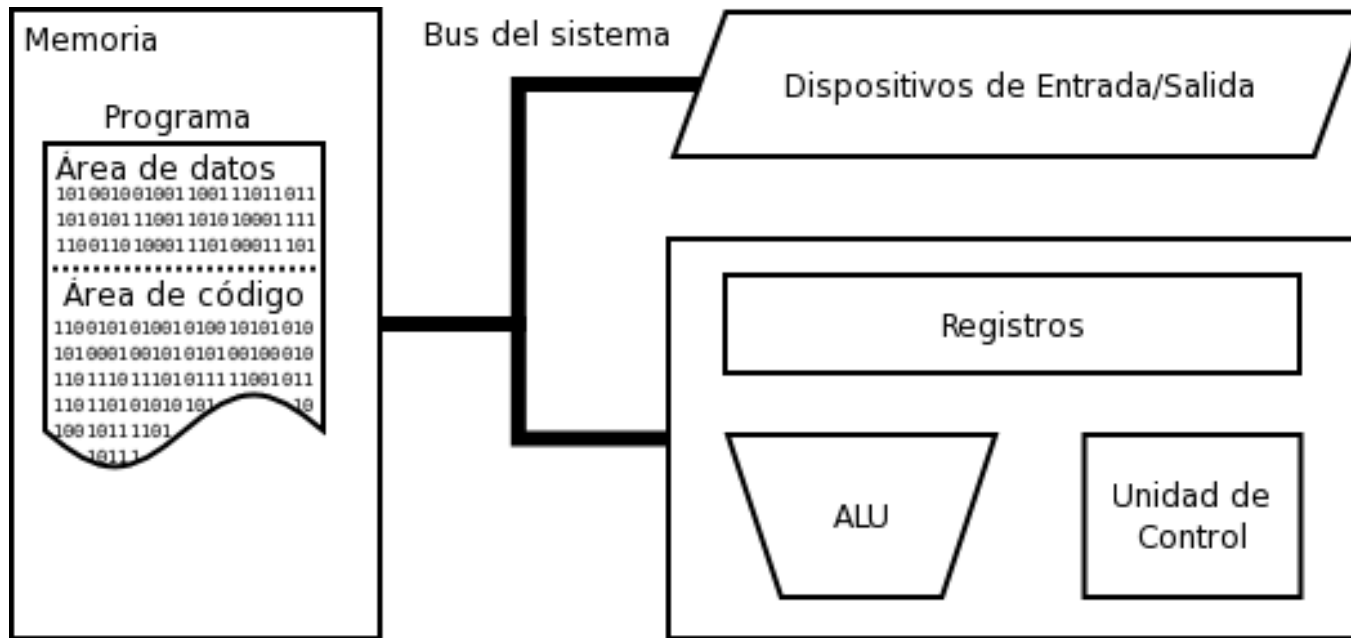


Consta de 5 componentes principales:

- Unidad de entrada: *provee las instrucciones y los datos.*
- Unidad de memoria: *donde se almacenan datos e instrucciones.*
- Unidad aritmetico-logica: *procesa los datos.*
- Unidad de control: *dirige la operación.*
- Unidad de salida: *se envían los resultados.*

Estructura – Modelo Von Neumann

Aspectos más importantes



• Utilización del sistema binario:

- Simplifica la implementación de funciones.
- Disminuye la probabilidad de fallos.

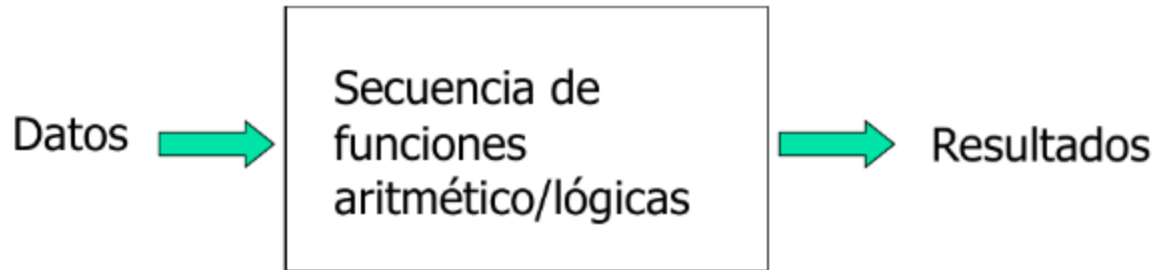
• Instrucciones y datos residen en memoria:

- Ejecución del programa en forma secuencial.
- Aumenta la velocidad.

• La memoria es direccionable por locación sin importar el dato almacenado.

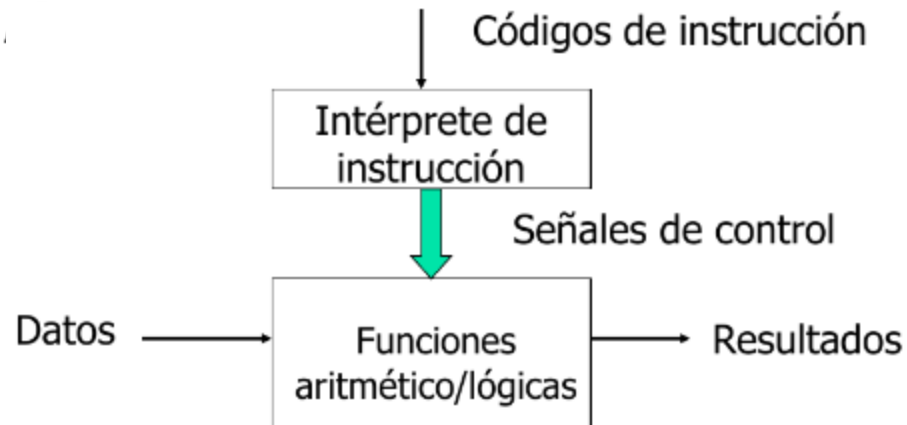
Concepto de Programa Pre y pos Von Neumann

Antes



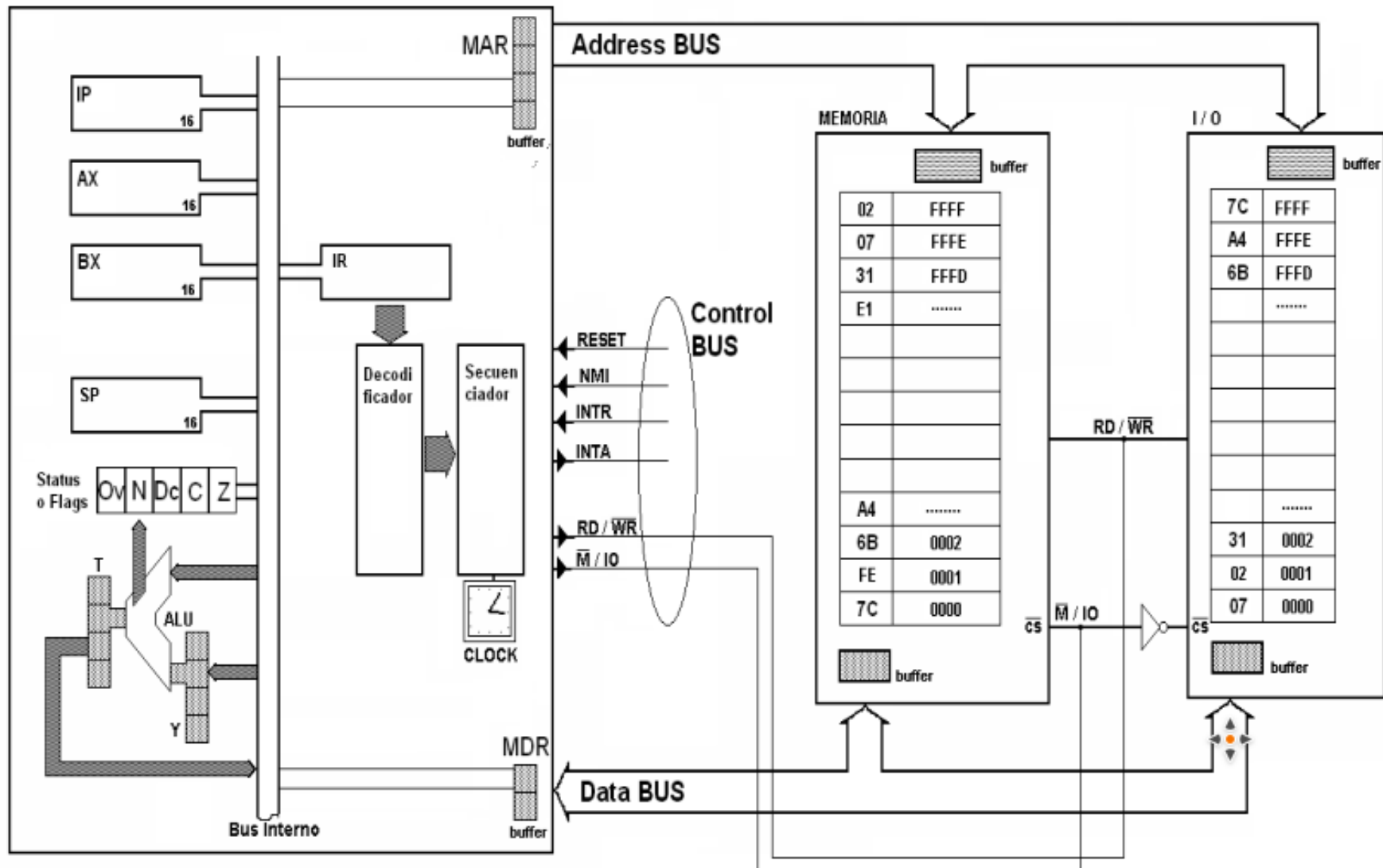
- Programación en hardware: cambio de tareas → se debe cambiar el hardware.

Ahora



- Programación en software: en c/ paso se efectúa alguna operación con los datos.
- Para cada paso se necesita un nuevo conjunto de señales de control.
- Las instrucciones proporcionan esas señales de control.
- Aparece un nuevo concepto de Programación. No hay que cambiar hardware.

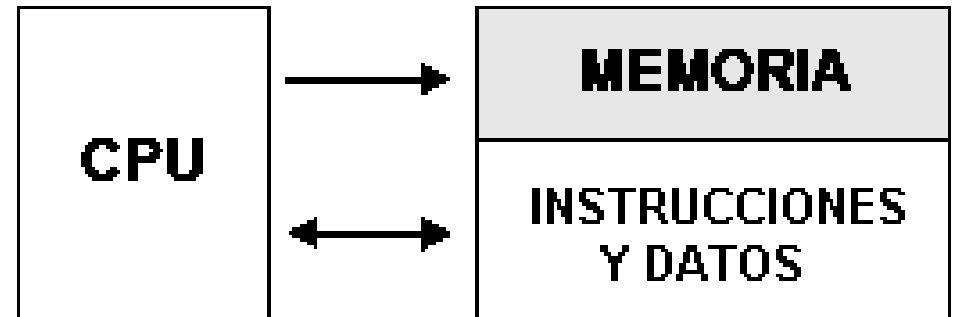
Arquitectura Interna Simplificada Von Neumann – Intel – mod. estudio



Arquitectura Von Neumann vs HARVARD

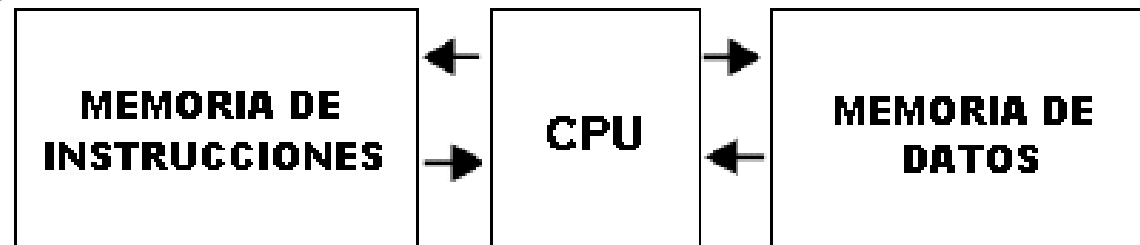
. Von Neumann.

- Estructura que posee un único BUS de direcciones, datos y control y la memoria se divide internamente en memoria de programa y de datos. No permite acceso simultáneo.

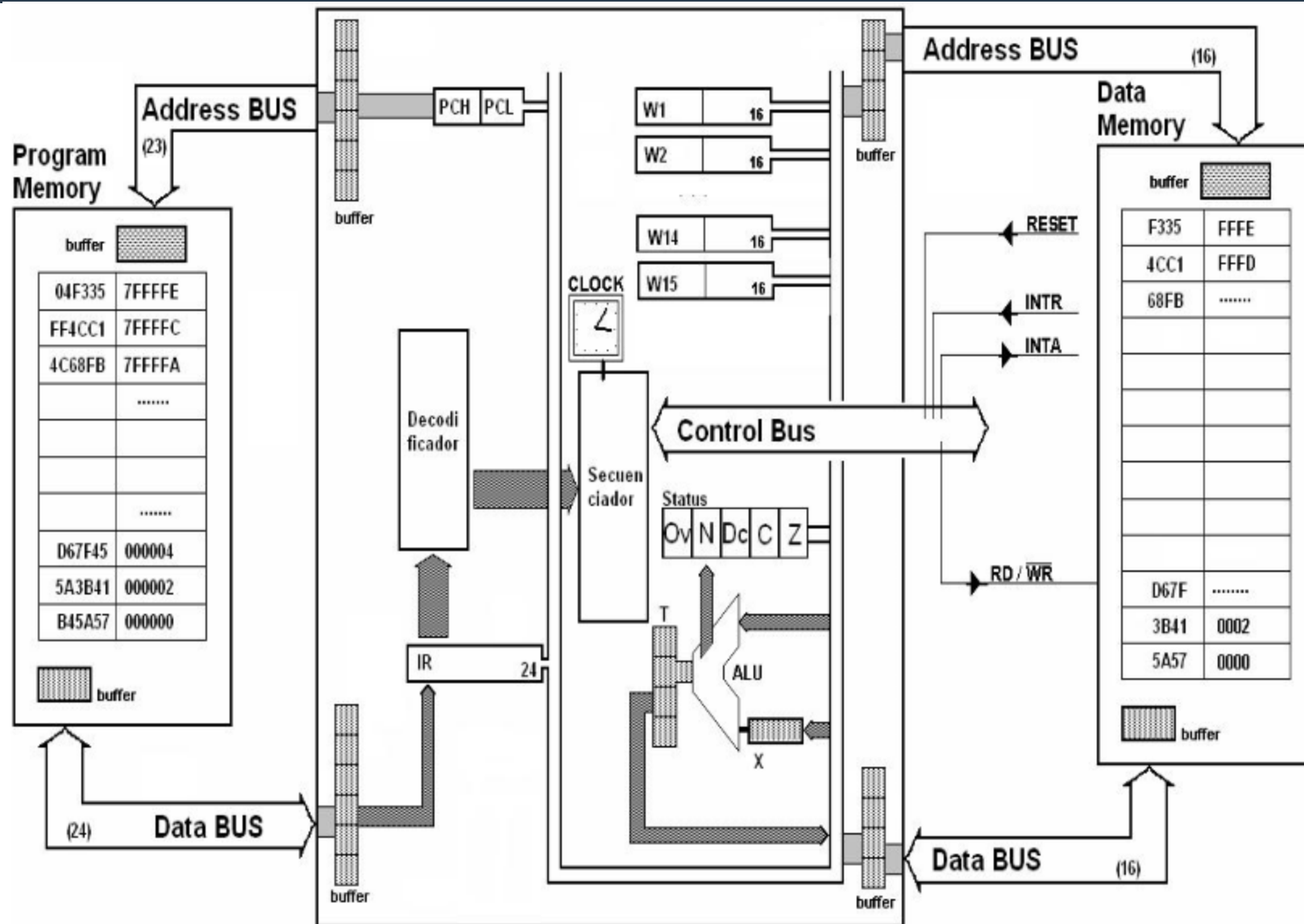


. Harvard.

- Estructura que posee dos BUSES de direcciones, datos y control y la memoria de programa y de datos son unidades separadas. De distinto tamaño de palabra. Permite acceso simultáneo.



Arquitectura Interna Simplificada HARVARD – Microchip 16 bits



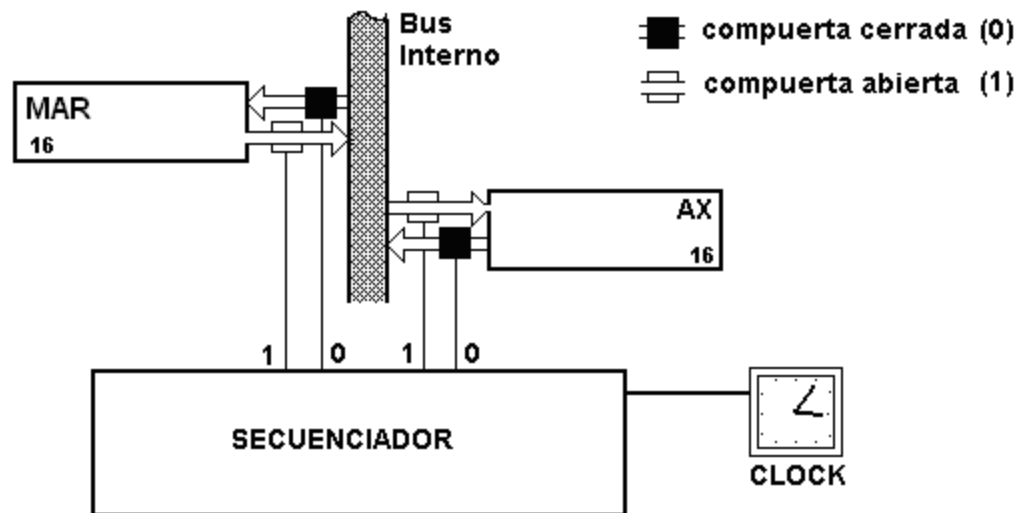
CPU – Bloques básicos

Transferencia entre registros

. Bloques básicos del CPU.

- Circuitos de tiempo y sincronismo (Clock y Secuenciador).
- El decodificador de Instrucciones.
- La ALU (Aritmethic – Logic Unit).
- Registros (W0...W15, PC, IR).
- Registros buffer (MAR, MDR, Adaptadores electrónicos del Bus).

. Transferencia entre registros:



Ciclos - Reloj

Formas de onda del Clock

. Clock Cycle - Ciclo de reloj.

Período "T" que media entre dos puntos homólogos de pasaje de la Onda. Segmento "A". Existen 2 flancos.

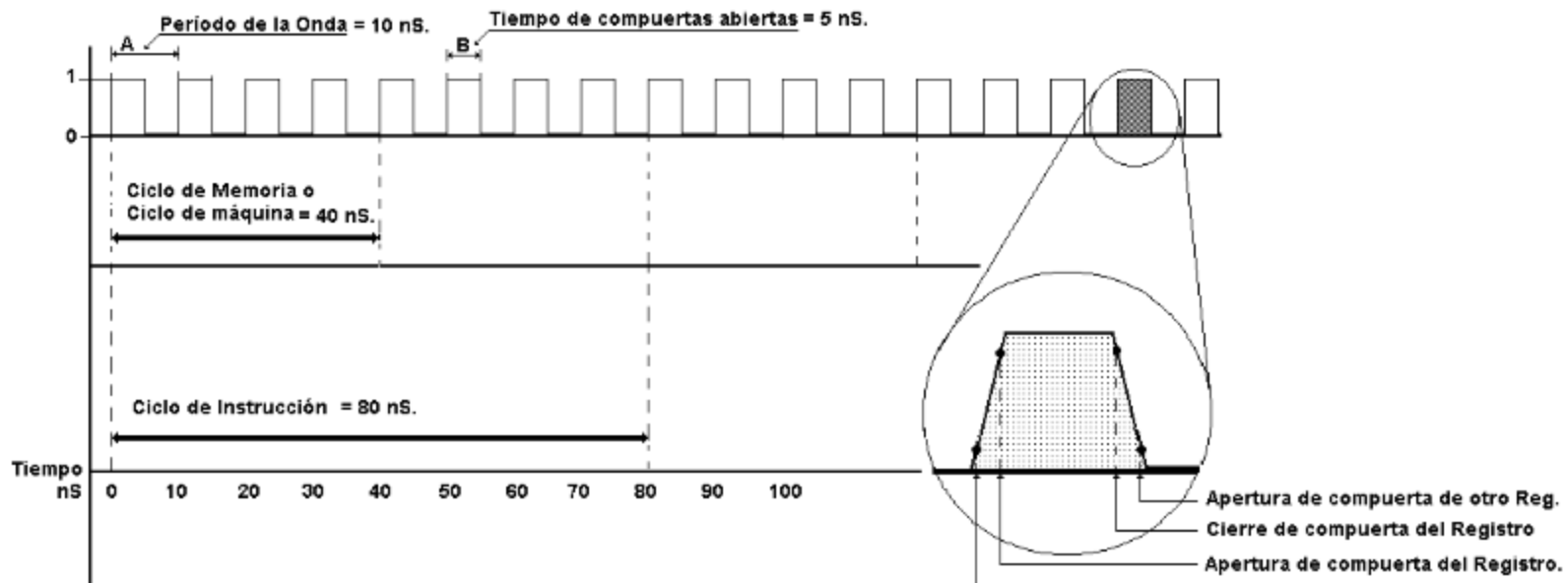
$$f = 1 / T. \quad T = 1 / 100\text{MHZ} = 1 / 100\text{M} * S^{-1} = 10 \text{ nS} = 10 \text{ nano seg.}$$

. MPU Cycle o Memory Cycle.

La memoria contiene un **gran número de compuertas**. Tiempo total de operación no es despreciable. Tiempo de Acceso a Memoria (**TA**), es el que media entre el instante en que se le suministra una dirección efectiva (**EA**) a la memoria en el **Address Bus**, hasta el instante en que el contenido de la locación especificada se encuentra en el **Data Bus**. El tiempo de acceso a memoria es el más importante (las velocidades dentro del micro son mucho mayores).

. Instruction Cycle – Ciclo de Instrucción.

Determinado por la cantidad de accesos a memoria que posee la instrucción en cuestión. Un Ciclo de instrucción puede comprender **uno o muchos ciclos de máquina**. Posee siempre dos estados fundamentales en los cuales se encuentra el uP, búsqueda (**FETCH**) o ejecución (**EXECUTE**).



Funcionamiento paso a paso CPU

Microinstrucciones

Un microprograma es una secuencia de datos binarios o microinstrucciones que representan señales eléctricas internas de la unidad de control de un microprocesador. Unas pocas de estas microinstrucciones implementan una instrucción completa del microprocesador.

Supongamos que se quiere ejecutar un programa, sumar dos valores de 16 bits, guardando el resultado en memoria. El programa cuenta con tres instrucciones. Su código Hexadecimal es el que figura a continuación (tabla en memoria a partir de la dirección 0200h).

Se utilizan sentencias de un microprocesador comercial del fabricante MICROCHIP

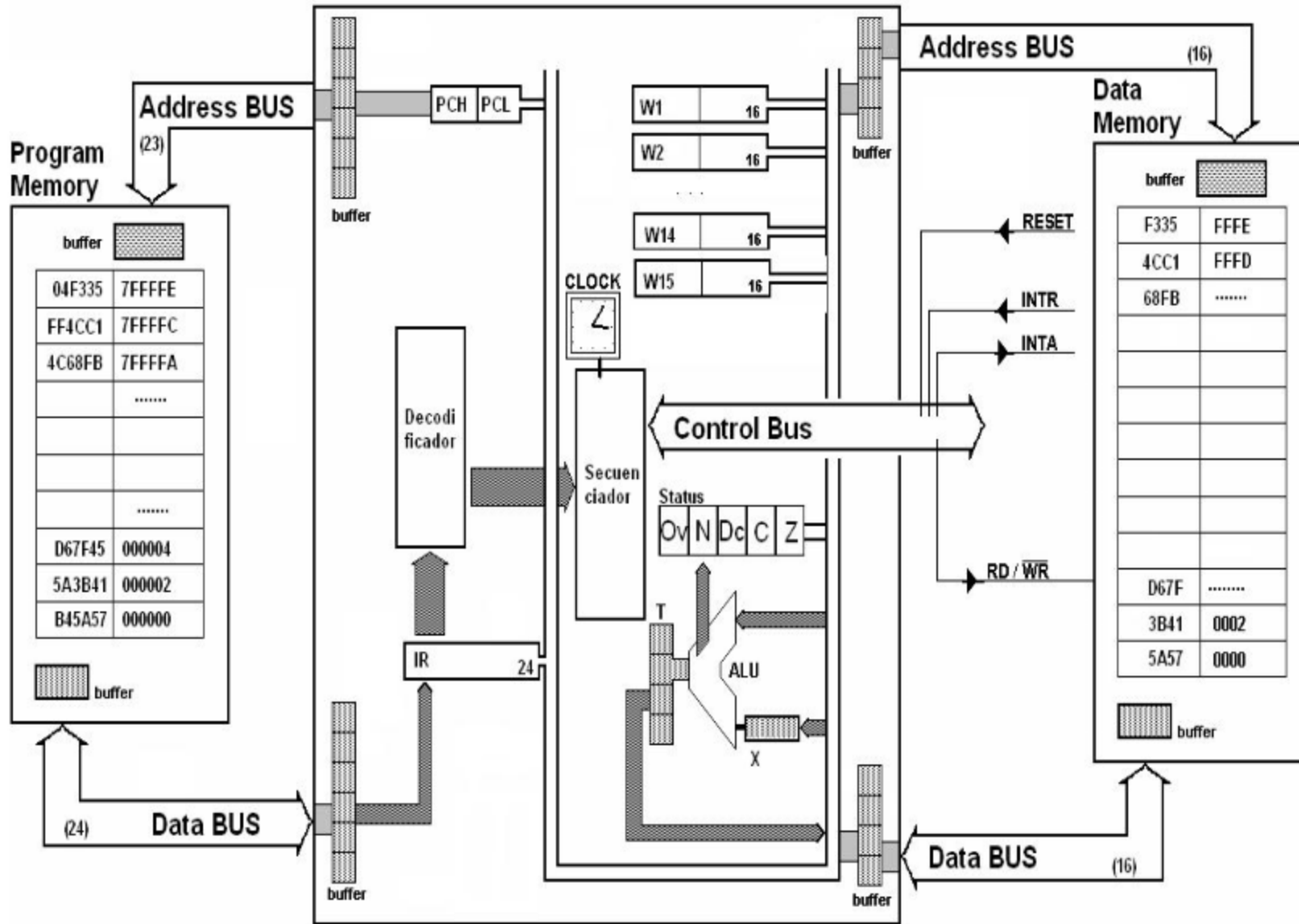
MEMORIA PROGRAMA

Dir.	Cont.
0200	00 B1
0201	00 20
0202	03 01
0203	00 40
0204	40 96
0205	00 88
0206	...
0207	...
0208	...

#	Assembler	Simbólico	Código
1.	MOV #0x000B, W1	000Bh→W1	20 00 B1
2.	ADD W0, W1, W6	W0 + W1→W6	40 03 01
3.	MOV W6, 0x0812	W6→(0812h)	88 40 96

Microinstrucciones (Ejemplo)

#1. MOV #0x000B, W1



1.

Words: 1

Cycles: 1

PC = 0200 → MAR → Address BUS → ProgMemory

1) $PC + 2 \rightarrow PC$

2) Retardo

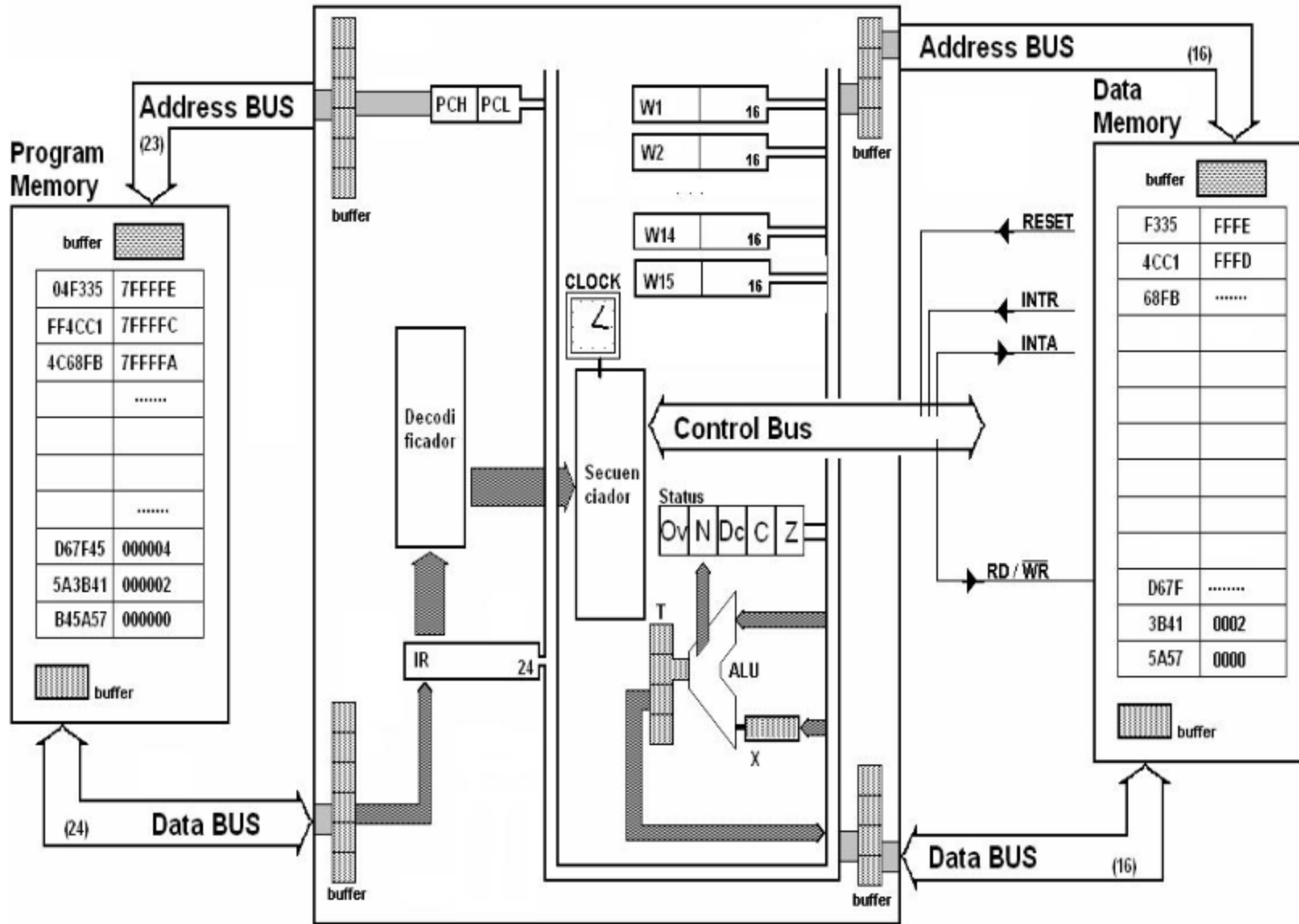
3) $[0200] = 2000B1 \rightarrow$ Data Bus → MDR

4) MDR → IR → Decodificador ⇒ `MOV #0x000B, W1`

5) $W1 = 0x000B$

Microinstrucciones (Ejemplo)

#2. ADD W0, W1, W6



2.

Words: 1

Cycles: 1

PC = 0202 → MAR → Address BUS → ProgMemory

1) PC + 2 → PC

2) Retardo

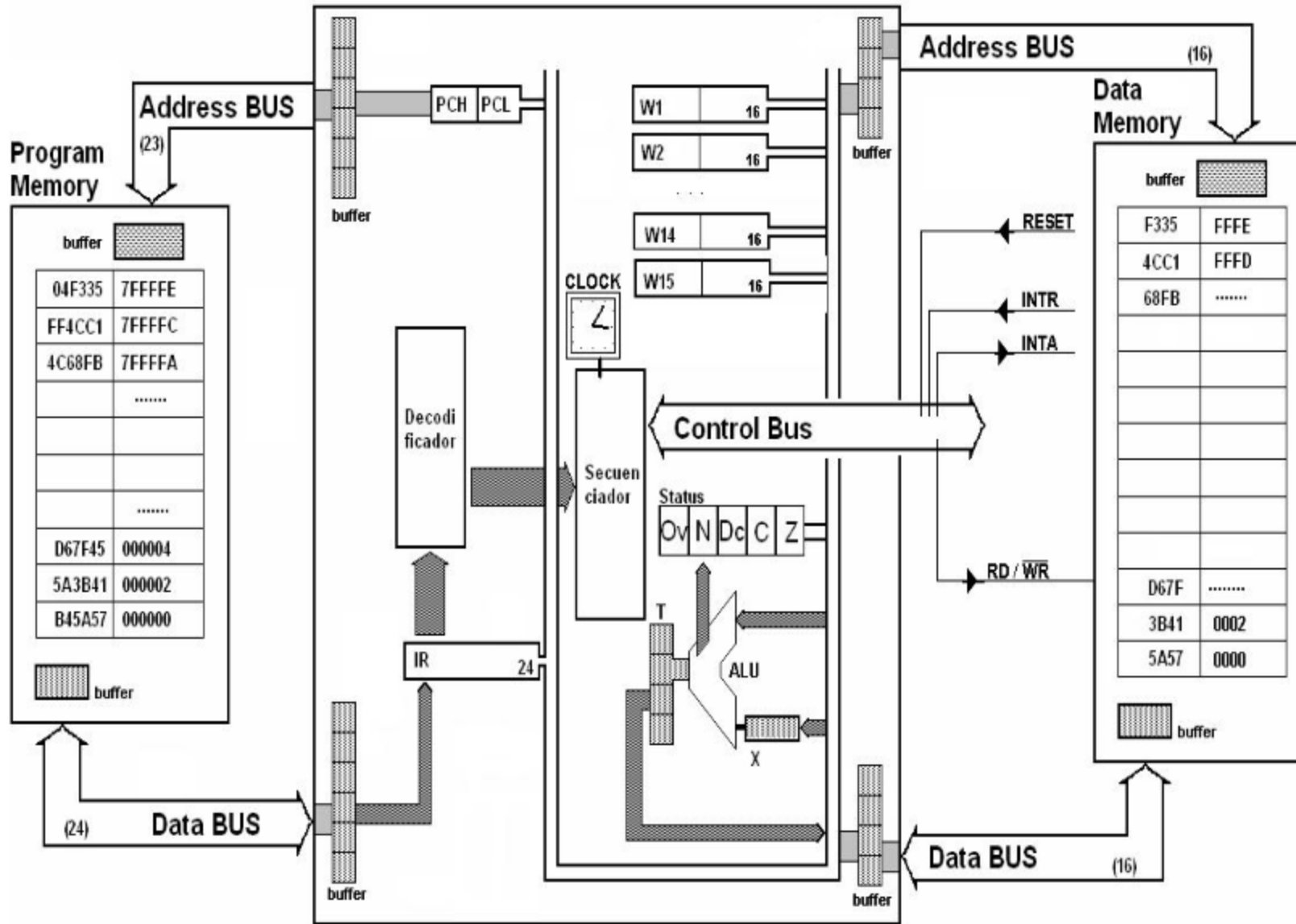
3) [0202] = 400301 → Data Bus → MDR

4) MDR → IR → Decodificador ⇒ MOV W0, W1, W6

5) W0 → X(ALU) W1 → ALU (ADD) → T(FLAGS) → W6

Microinstrucciones (Ejemplo)

#3. MOV W6, 0x0812



3.

Words: 1

Cycles: 1

PC = 0204 → MAR → Address BUS → ProgMemory

1) $PC + 2 \rightarrow PC$

2) Retardo

3) $[0204] = 884096 \rightarrow$ Data Bus → MDR

4) MDR → IR → Decodificador ⇒ `MOV W6, 0x0812`

5) $MAR = 0812 \rightarrow$ Address BUS → DataMemory

6) $W6 \rightarrow$ MDR → DataBus → `[0812]`

Microinstrucciones (Ejemplo)

Entorno de desarrollo (MPLABX)

```
1  ;Declaración de Archivos a incluir y EQUs
2  .include "p33FJ256GP710.inc"
3  .global _main
4
5  .section .data
6  ;Inicializar variables de 16 bits en memoria
7
8  .section .text
9
10 _main:
11     MOV #0x000B, W1
12     ADD W0, W1, W6
13     MOV W6, 0x0812
14     .end                                ;EOF
15
```

Output Variables Call Stack Breakpoints Program Memory X

	Line	Address	Opcode	Label	DisAssy
	357	002C8	DAT000		BREAK
	358	002CA	FE0000		RESET
	359	002CC	050000		RETLW #0x0, W0
➡	360	002CE	2000B1		MOV #0xB, W1
	361	002D0	400301		ADD W0, W1, W6
	362	002D2	884096		MOV W6, 0x812
	363	002D4	FFFFFF		NOPR

Preguntas...???

GRACIAS!!!