

# Tema 3: Introducción a la Arquitectura Von Neumann

- 3.1- Máquinas específicas y máquinas genéricas
- 3.2- Unidades funcionales y principios de funcionamiento



## 3.1- Máquinas específicas y máquinas genéricas

### RESOLUCIÓN DE PROBLEMAS

#### Planteamiento



#### Algoritmo

Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema

#### Ejemplo

Hallar el máximo de una lista de números



### 3.1- Máquinas específicas y máquinas genéricas

1. **Problema** → Hallar el máximo de una lista de números

2. **Algoritmo** →

- Inicializar **máximo**  $\leftarrow 0$
- Para cada **número** de la lista, hacer:
  - Comparar **número** con **máximo**
  - Si **número** > **máximo**  
asignar **máximo**  $\leftarrow$  **número**
  - Si no  
asignar **máximo**  $\leftarrow$  **máximo**

3. **Solución**

Máquina específica  
Máquina genérica



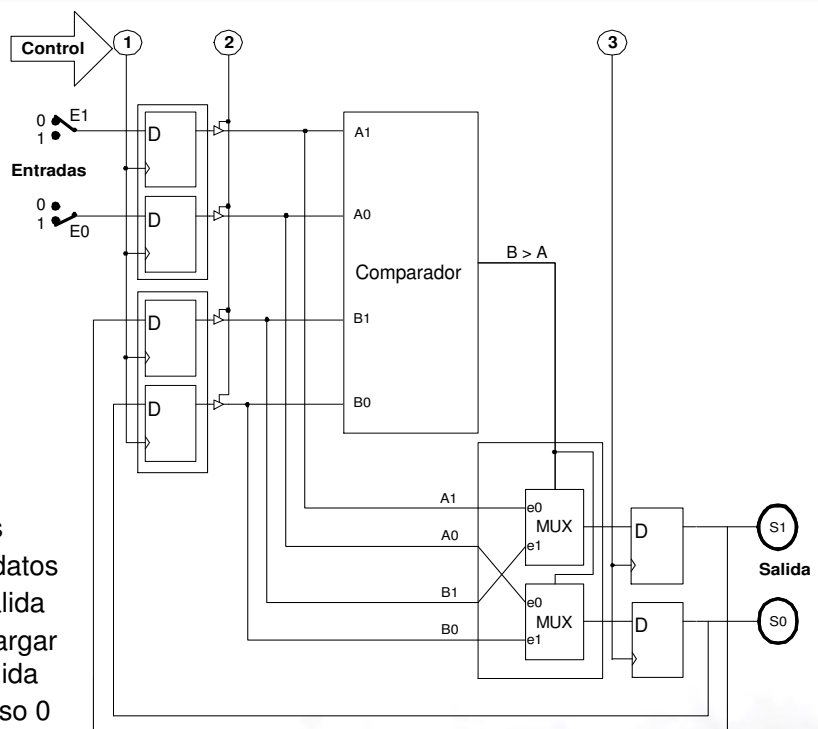
### 3.1- Máquinas específicas y máquinas genéricas

#### Máquina Específica

A1	A0	B1	B0	B > A
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

#### Pasos:

0. Introducir datos en Entradas
1. Pulsar botón 1 para cargar datos
2. Pulsar 2 hasta generar la salida
3. Pulsando 2, pulsar 3 para cargar máximo en el registro de salida
4. Soltar botón 2 y volver al paso 0



## 3.1- Máquinas específicas y máquinas genéricas

### Máquina Genérica

#### ■ Objetivo

Construir una máquina genérica que pueda ejecutar cualquier algoritmo

#### ■ Necesidades → Solucionadas con tecnología digital

1. Almacenar el algoritmo para poder ser modificado o cambiado por otros → **Memorias digitales**
2. Almacenar los datos → **Memorias digitales**
3. Un mecanismo que lea el algoritmo desde su almacén y lo “ejecute” → **CPU (Central Processing Unit)**
4. Un mecanismo que permita introducir los datos y los algoritmos en la máquina y extraer los resultados  
→ **Dispositivos de Entrada/Salida y periféricos**

Los datos y los algoritmos deben poder codificarse como secuencias de bits para ser almacenados en memorias digitales



## 3.1- Máquinas específicas y máquinas genéricas

### Concepto de instrucción

- **Instrucción:** Operación que la CPU sabe hacer porque la lleva cableada en su diseño.

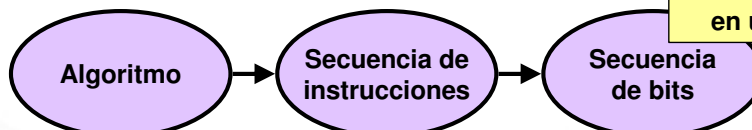
Ejemplo: Sumar dos cantidades de 16 bits

- El conjunto de instrucciones que una CPU sabe hacer por diseño es **finito**

### Codificación del algoritmo

- El algoritmo debe ser reformulado para cada CPU como una **secuencia de instrucciones** que sabe esa CPU
- Cada posible instrucción de la CPU tiene un código binario (secuencia de bits) diferente

El algoritmo, como los datos, se puede almacenar en una memoria digital



### Arquitectura Von Neumann

