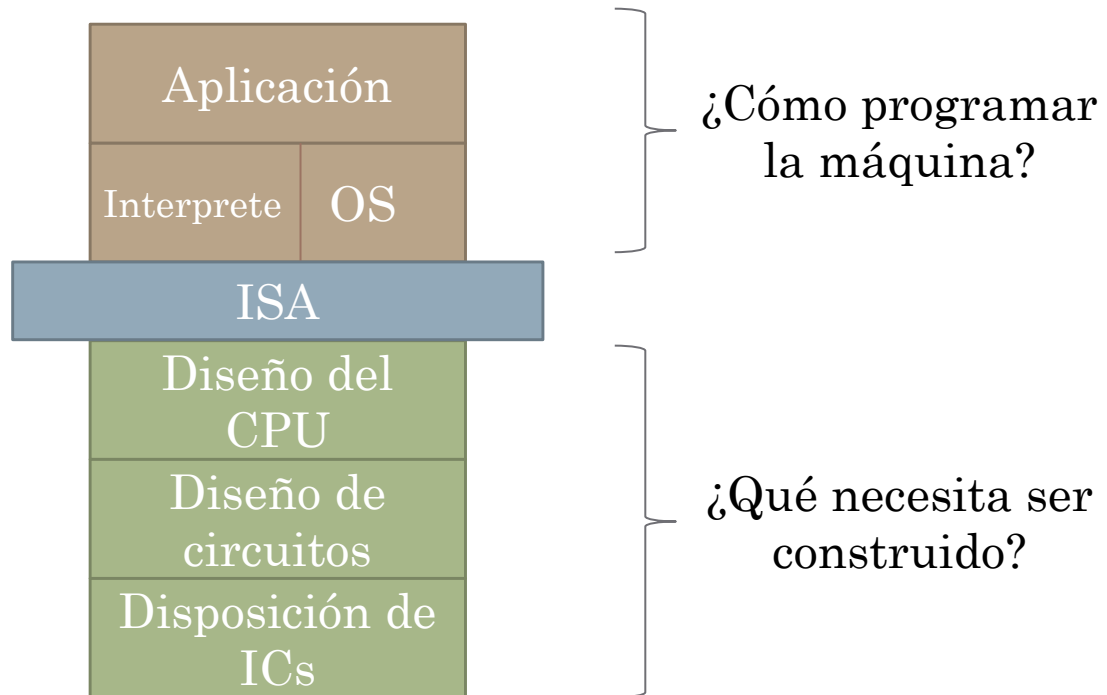


Arquitectura de computadoras

- Atributos de un sistema visibles al programador (tienen un impacto directo en la ejecución lógica de un programa).
- **Arquitectura de conjunto de instrucciones** (Instruction Set Architecture, ISA).

8086/8088 Instruction Set Summary

Mnemonic and Description	Instruction Code			
DATA TRANSFER				
MOV = Move:	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Register/Memory to/from Register	1 0 0 0 1 0 d w	mod reg r/m		
Immediate to Register/Memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate to Register	1 0 1 1 w reg	data	data if w = 1	
Memory to Accumulator	1 0 1 0 0 0 w	addr-low	addr-high	
Accumulator to Memory	1 0 1 0 0 1 w	addr-low	addr-high	
Register/Memory to Segment Register	1 0 0 0 1 1 1 0	mod 0 reg r/m		
Segment Register to Register/Memory	1 0 0 0 1 1 0 0	mod 0 reg r/m		
PUSH = Push:				
Register/Memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m		
Register	0 1 0 1 0 reg			
Segment Register	0 0 0 reg 1 1 0			
POP = Pop:				
Register/Memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m		
Register	0 1 0 1 1 reg			
Segment Register	0 0 0 reg 1 1 1			

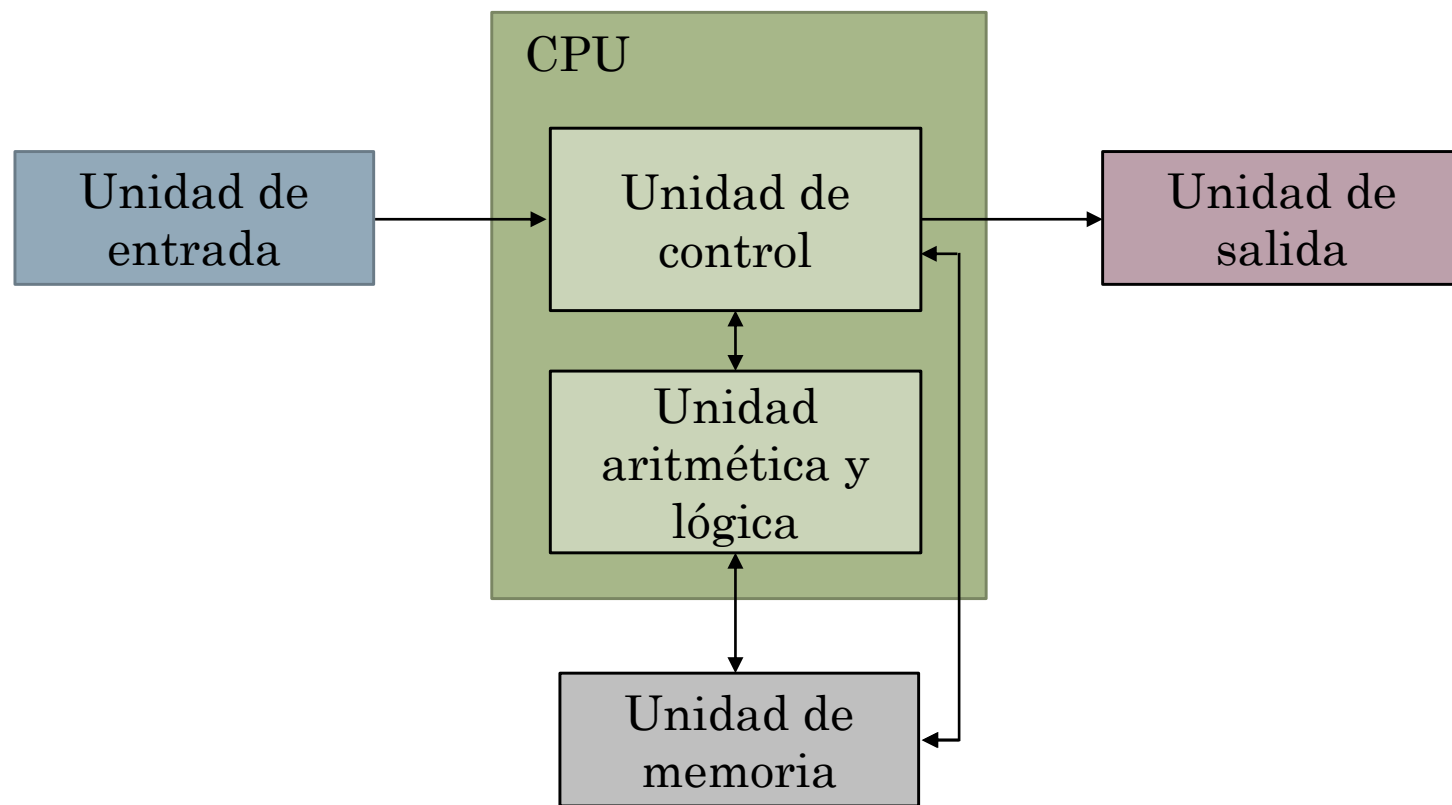


Organización de computadoras

Se refiere a las unidades operativas y sus interconexiones que realizan las especificaciones arquitectónicas.

Ejemplos de atributos **arquitectónicos** son el conjunto de instrucciones, el número de bits usados para representar tipos de datos, mecanismos de E/S y técnicas para direccionar memoria.

Los atributos **organizacionales** incluyen aquellos detalles de hardware transparentes al programador, tales como señales de control, interfaces entre la computadora y los periféricos y la tecnología de memoria usada.



Ejemplo

- Es una cuestión de **diseño arquitectónico** si una computadora tendrá una instrucción de multiplicación.
- Es una cuestión **organizativa** si la instrucción va a ser implementada por una unidad de multiplicación especial o por un mecanismo que usa repetidamente la unidad de suma del sistema.

- Muchos fabricantes de computadoras ofrecen una familia de modelos de computadora, todos con la misma arquitectura pero con diferencias en la organización.
- En consecuencia, los diferentes modelos de la familia tienen **diferentes características de precio y rendimiento**. Además, una arquitectura en particular puede abarcar muchos años y comprender varios modelos de computadora diferentes, y su organización cambia con la tecnología cambiante.

Estructura y función de una computadora

- **Estructura:** la forma en que los componentes están interrelacionados en un sistema jerárquico.
- **Función:** el funcionamiento de cada componente individual como parte de la estructura.

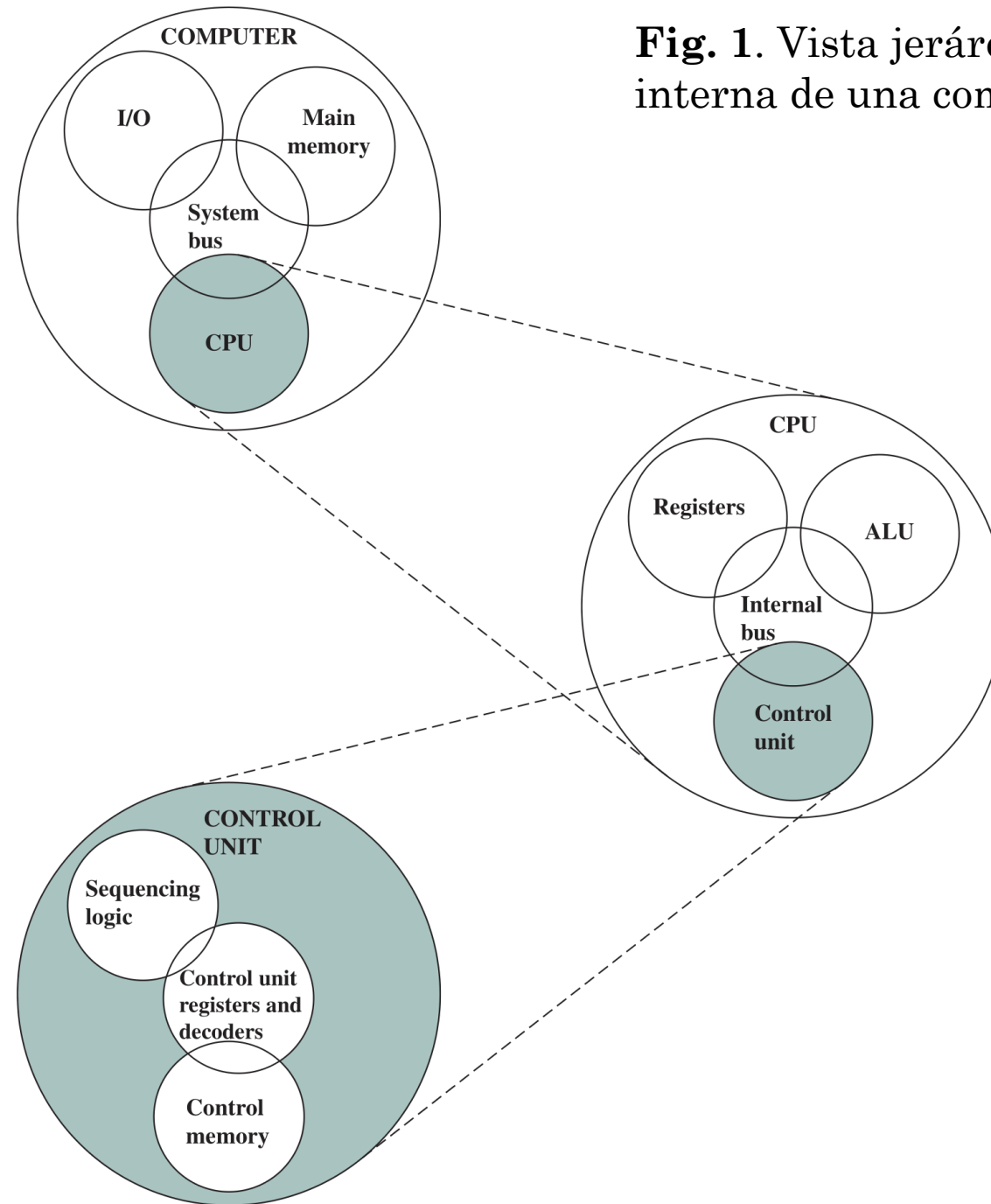
Función de una computadora

- En términos generales, son cuatro las funciones básicas que una computadora puede realizar:
- **Procesamiento de datos.**
- **Almacenamiento de datos.**
- **Movimiento de datos.**
- **Control.**

Estructura de una computadora

- Hay cuatro componentes estructurales principales en una computadora tradicional:
- **Unidad central de procesamiento (CPU).**
- **Memoria principal.**
- **E/S.**
- **Interconexión del sistema.**

Fig. 1. Vista jerárquica de la estructura interna de una computadora tradicional



Rendimiento de una computadora

Entre las técnicas integradas en los procesadores contemporáneos para incrementar el rendimiento de los procesadores se encuentran las siguientes.

- **Pipelining (segmentación encauzada)**

Instr. 1	TI	DI	CO	TO	EI	ER			
Instr. 2		TI	DI	CO	TO	EI	ER		
Instr. 3			TI	DI	CO	TO	EI	ER	
Instr. 4				TI	DI	CO	TO	EI	ER

TI: Traer instrucción.
DI: Decodificar instrucción.
CO: Calcular operandos.
TO: Traer operandos.
EI: Ejecutar instrucción.
ER: Escribir resultado.

- **Predicción de bifurcaciones**

```
if(a == 5)
    a = 0;
if(b == 5)
    b = 0;
if(a != b)
    // algunas acciones
else
    // otras acciones
```

- **Ejecución superescalar.**

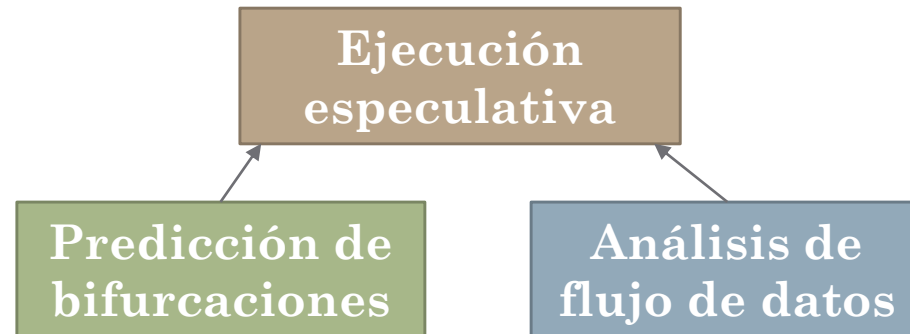
Instr. 1	TI	DI	CO	TO	EI	ER	
Instr. 2	TI	DI	CO	TO	EI	ER	
Instr. 3		TI	DI	CO	TO	EI	ER
Instr. 4		TI	DI	CO	TO	EI	ER

TI: Traer instrucción.
DI: Decodificar instrucción.
CO: Calcular operandos.
TO: Traer operandos.
EI: Ejecutar instrucción.
ER: Escribir resultado.

- **Análisis de flujo de datos.**

```
a = Mem[10]  
a = a + 1  
Mem[20] = a  
a = Mem[30]  
a = a + 2  
Mem[40] = a
```

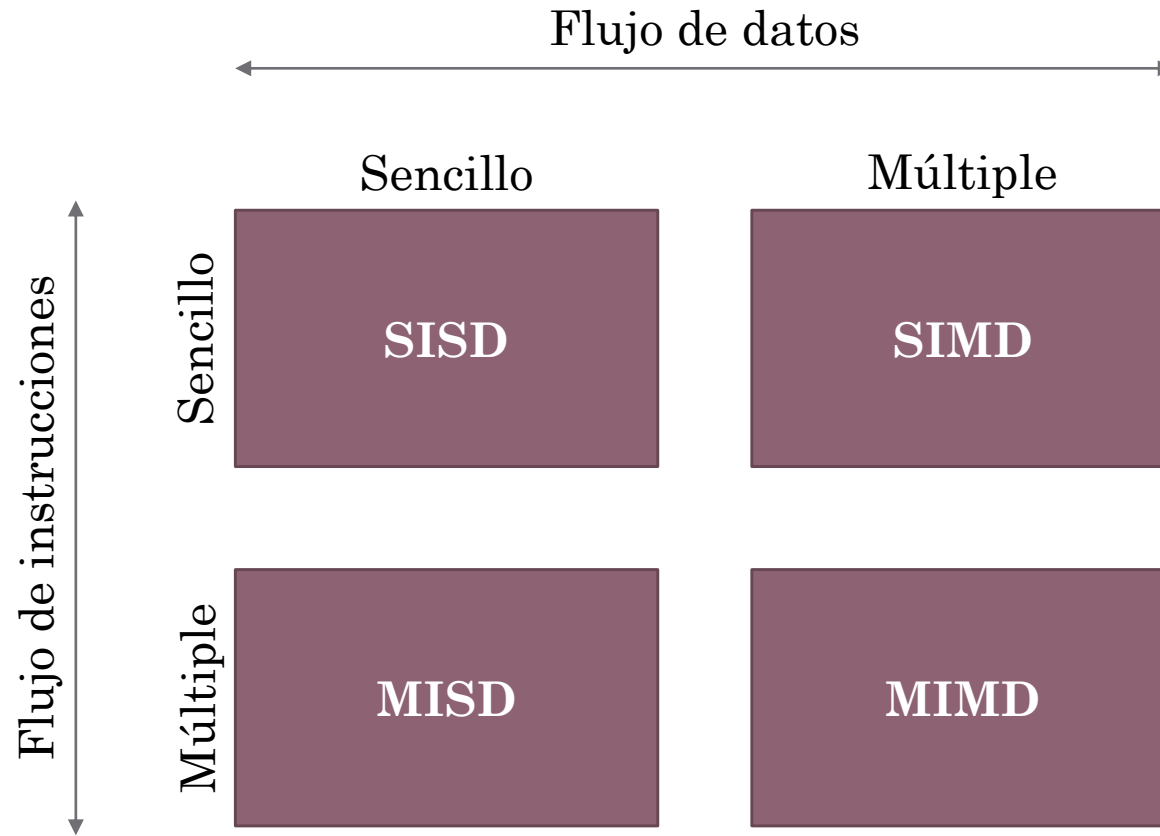
- **Ejecución especulativa.**



Taxonomía Flynn

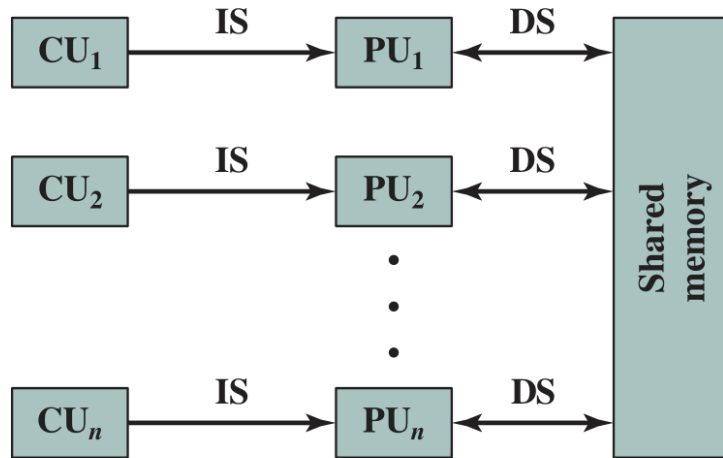
La taxonomía más aceptada hoy en día para categorizar arquitecturas de computadora es la propuesta por Michael Flynn en 1972. La taxonomía de Flynn considera dos factores: el número de flujos de instrucciones y de datos que fluyen hacia el procesador.

- **SISD** (single instruction stream, single data stream).
- **SIMD** (single instruction stream, multiple data streams).
- **MISD** (multiple instruction streams, single data stream).
- **MIMD** (multiple instruction streams, multiple data streams)

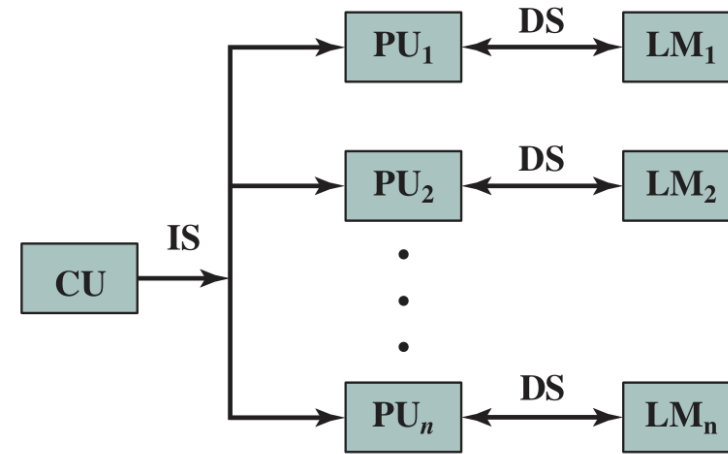




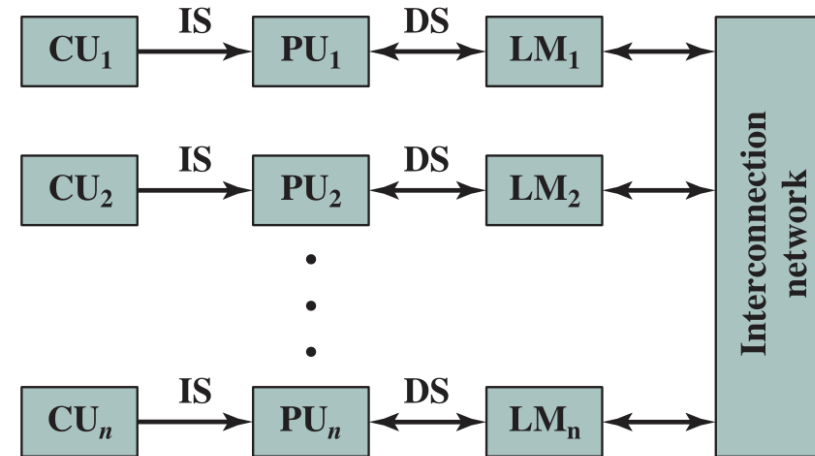
(a) SISD



(c) MIMD (with shared memory)



(b) SIMD (with distributed memory)



(d) MIMD (with distributed memory)

CU = Control unit
 IS = Instruction stream
 PU = Processing unit
 DS = Data stream
 MU = Memory unit
 LM = Local memory

SISD = Single instruction,
 = single data stream
 SIMD = Single instruction,
 multiple data stream
 MIMD = Multiple instruction,
 multiple data stream