



# ***Introducción a los sistemas digitales***

© Luis Entrena, Celia López, Mario García,  
Enrique San Millán

Universidad Carlos III de Madrid

# Índice

---

- Diseño de sistemas digitales
  - Proceso de diseño de sistemas digitales
  - Estructura de un sistema digital
    - La ruta de datos
    - La unidad de control
- Circuitos integrados
  - Tecnologías digitales
  - ASICs
  - Dispositivos programables
  - Microprocesadores

# Sistemas digitales

---

- Los sistemas digitales procesan información digital, de acuerdo con un algoritmo determinado
- Algoritmo: conjunto ordenado y finito de operaciones que permite hallar la solución a un problema
- Objetivo de la lección: introducción a los sistemas digitales

# Diseño de sistemas digitales

---

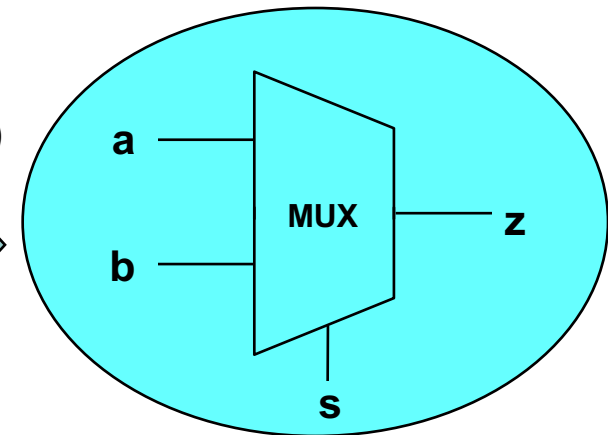
- El diseño con componentes tan básicos como las puertas lógicas no es práctico para diseñar sistemas digitales
- Diseño en el Nivel de Transferencia entre Registros (Register Transfer Level, RTL):
  - Componentes más abstractos: ALUs, registros o multiplexores
  - El sistema se describe como operaciones que se realizan entre datos almacenados en registros

# Lenguajes de Descripción de Hardware

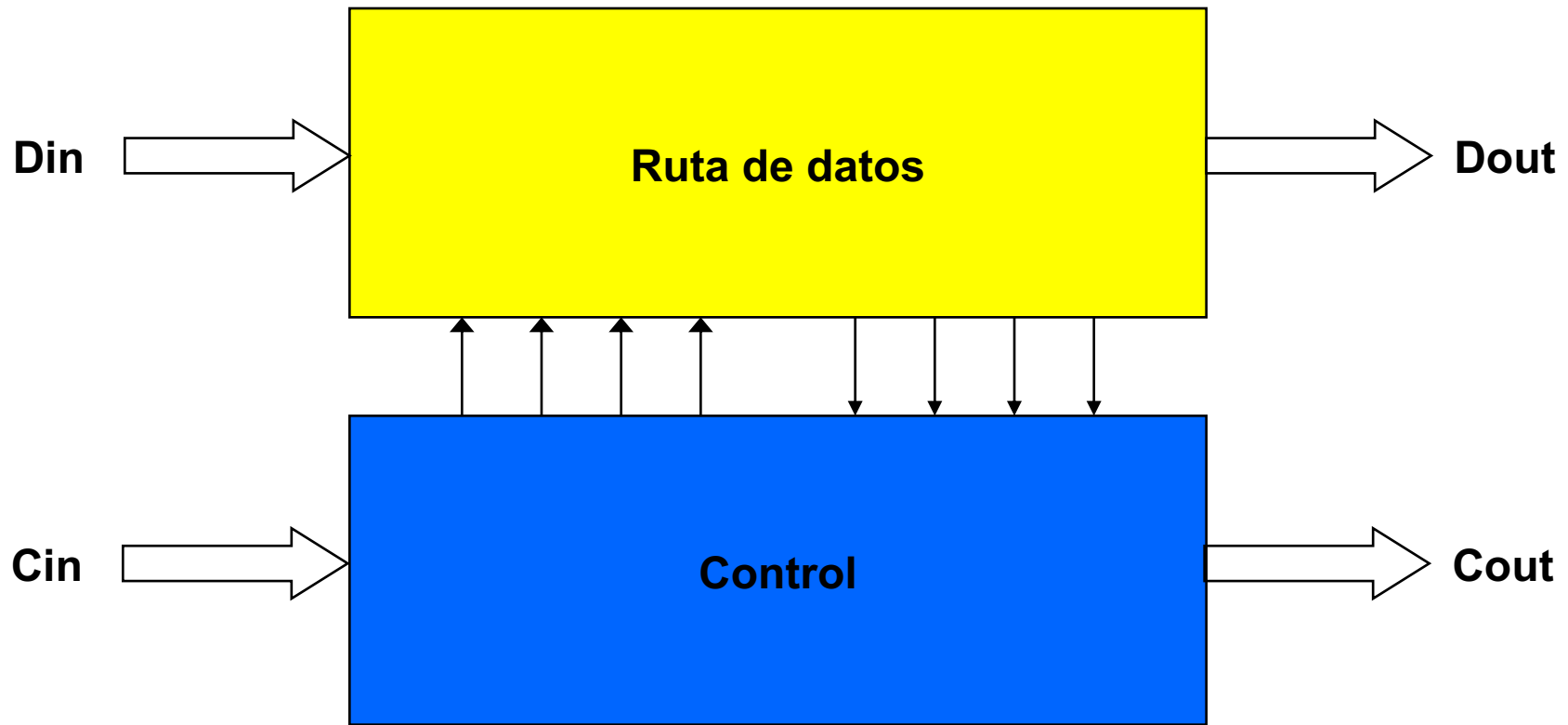
- Los Lenguajes de Descripción de Hardware o HDLs (Hardware Description Languages) permiten diseñar un circuito digital desde un mayor nivel de abstracción
- Ejemplo (VHDL)

```
if s = '0' then
    z <= a;
else
    z <= b;
end if;
```

Sintetizador  
(Compilador)



# Estructura de un sistema digital



# La ruta de datos

---

- Todo algoritmo se descompone en una serie de operaciones básicas:
  - Aritméticas
  - Lógicas
  - Desplazamientos y rotaciones
- La ruta de datos (“datapath”) es el conjunto de unidades funcionales que procesan los datos

# La ruta de datos

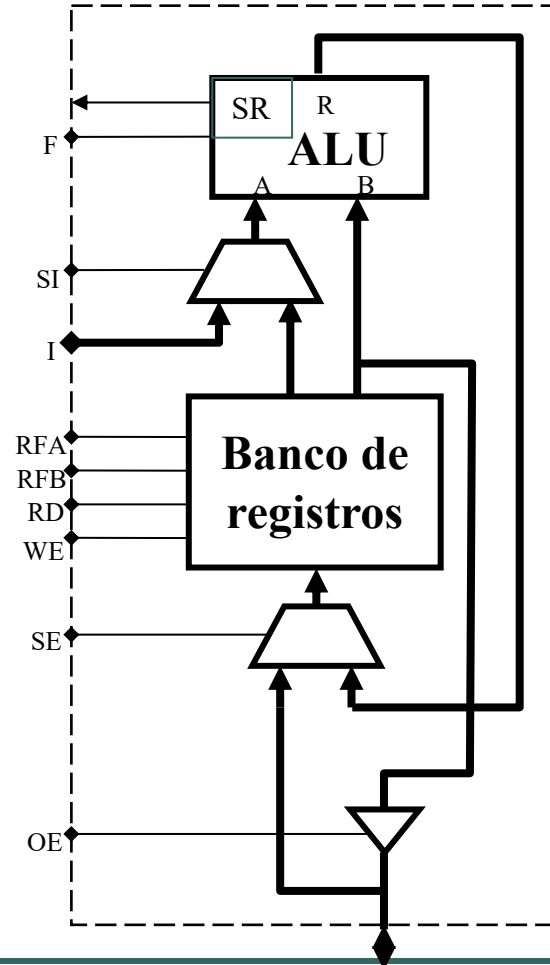
---

- Componentes típicos de la ruta de datos
  - ALUs: realizan las operaciones necesarias
  - Registros y memorias: almacenan datos temporales
  - Buses: conectan los elementos de la ruta de datos
  - Multiplexores: seleccionan los datos que se deben procesar en cada momento



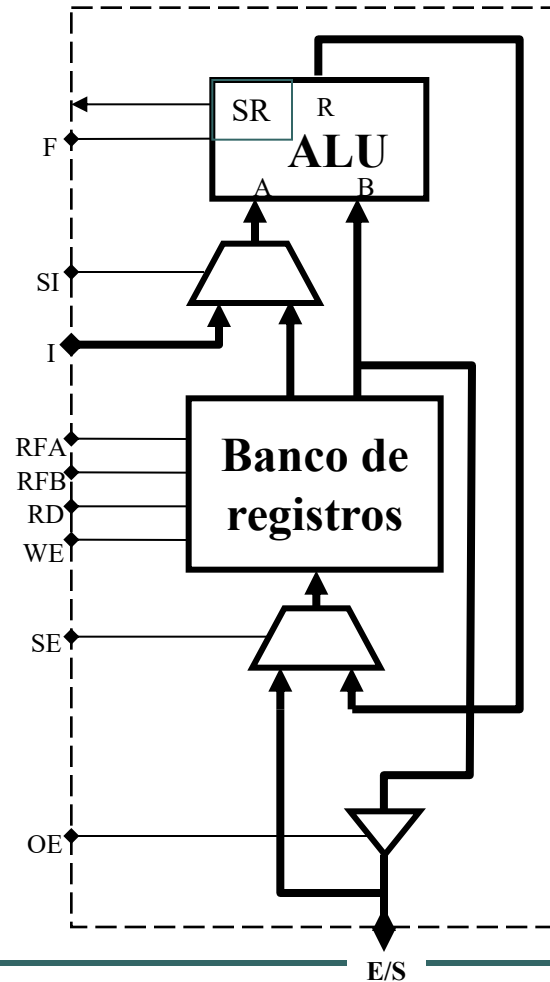
# Ejemplo de ruta de datos

- ALU realiza operaciones
  - F: Función de la ALU
- SR: Registro de estado. Indicadores (flags) de la operación realizada. Ejemplos:
  - C: Acarreo
  - O: Overflow
  - Z: Cero
  - S: Signo
- Operandos inmediatos a través de I, seleccionados con SI



# Ejemplo de ruta de datos

- Banco de registros (memoria de triple puerto) almacena valores intermedios
- Selección de registros
  - RFA: Registro Fuente A
  - RFB: Registro Fuente B
  - RD: Registro Destino
- E/S para conexión externa
  - SE selecciona la fuente para RD (externa o interna)
  - OE habilita la salida



# Unidad de control

---

- Determina la correcta secuenciación y utilización de las operaciones sobre los datos
- Elementos típicos:
  - Máquinas de estados
  - Contadores
  - Registros y memorias con datos de control, etc.

# Unidad de control

---

- Ejemplo: realizar la operación siguiente

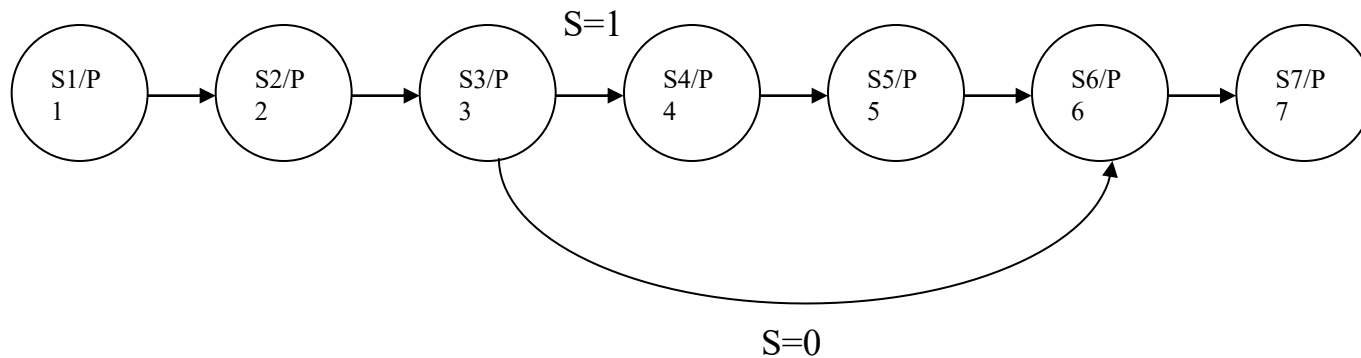
$$y = \text{abs}(x1 - x2)/2$$

- Pasos:

- 1. Cargar x1 en el registro R1
- 2. Cargar x2 en el registro R2
- 3. Restar x2 de x1 y colocar el resultado en el registro R3 ( $R3 = R1 - R2$ ).
  - Si el resultado es positivo, saltar al paso 6. En caso contrario, seguir con el paso 4.
- 4. Complementar R3 ( $R3 = \text{NOT } R3$ )
- 5. Incrementar R3 ( $R3 = R3 + 1$ )
- 6. Desplazar R3 a la derecha ( $R3 = R3/2$ )
- 7. Enviar R3 a la salida

# Unidad de control

- Máquina de Estados de la unidad de control



# Unidad de control

- Salidas de la Máquina de Estados

Paso	Función ALU (F)	RFA	SI	I	RFB	RD	WE	SE	OE
1	X	X	X	X	X	R1	0	1	0
2	X	X	X	X	X	R2	0	1	0
3	RESTA	R1	0	X	R2	R3	0	0	0
4	NOT	R3	0	X	X	R3	0	0	0
5	SUMA	X	1	1	R3	R3	0	0	0
6	SHR	R3	0	X	X	R3	0	0	0
7	X	X	X	X	R3	X	1	X	1

# Circuitos Integrados

---

- Componentes discretos (estándar)
  - 74xx, 54xx
- Circuitos integrados a medida
  - ASIC: Application Specific Integrated circuit
- Circuitos programables
  - PLD, SPLD: (Simple) Programmable Logic Devices
  - CPLD: Complex Programmable Logic Devices
  - FPGA: Field Programmable Gate Array
- Microprocesadores

# Circuitos Integrados (cont.)

---

- Sistemas microprocesadores
  - Microprocesador y componentes adicionales (componentes estándar)
  - SoC: System on Chip (ASIC)
  - SoPC: System on Programmable Chip (FPGA)



# Tecnologías digitales

---

- Las puertas lógicas son circuitos electrónicos
- El nivel lógico (0 o 1) se representa mediante un nivel de tensión
- Generalmente se utiliza “lógica positiva”
  - Tensión alta (5V, 3.3V, 2.5 V, etc)  $\rightarrow$  1
  - Tensión baja (0V)  $\rightarrow$  0
- Existen muchas tecnologías, según la forma en que se realizan las puertas lógicas y las características que se obtienen

# Características de las tecnologías digitales

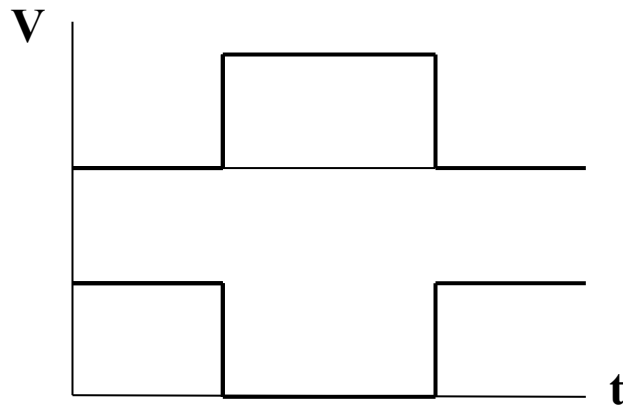
---

- Principales características:
  - Margen de temperaturas de operación
  - Tensión de alimentación
  - Margen de ruido (intervalos de tensiones que se asocian a un nivel lógico determinado)
  - Retardo de conmutación
  - Consumo
  - Otros
- Cada tecnología o subfamilia presenta valores diferentes respecto a estos parámetros

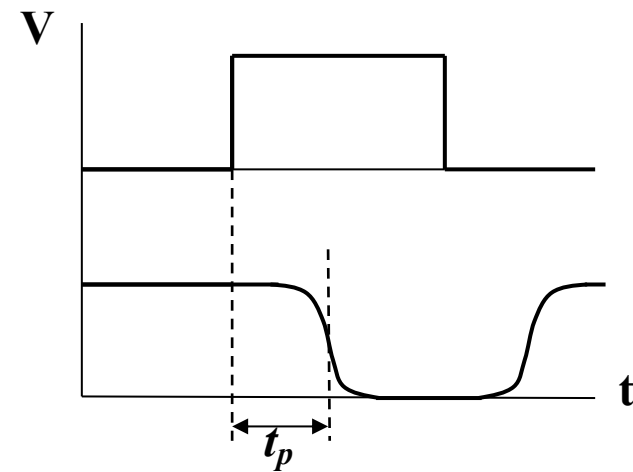
# Retardos

- Las puertas lógicas no conmutan instantáneamente

**Inversor ideal**



**Inversor real**



- El retardo limita la velocidad de operación del circuito

# Consumo

---

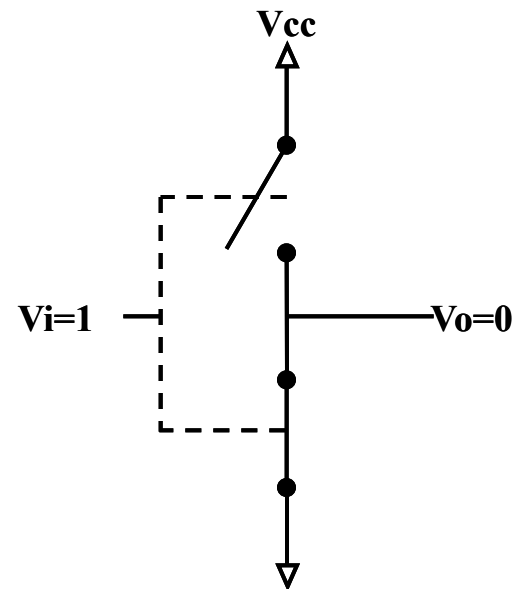
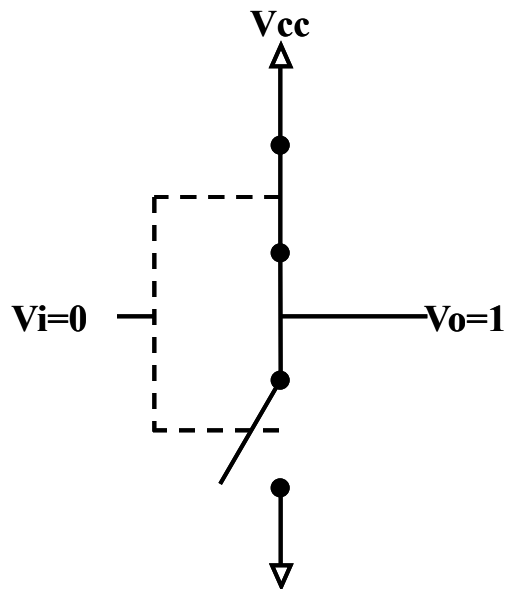
- Las puertas lógicas consumen energía:
  - Estática: la que se consume por tener alimentada la puerta lógica, sin cambiar los valores lógicos
  - Dinámica: la que se consume al conmutar
- En la tecnología CMOS (la más utilizada actualmente), el consumo estático es muy pequeño. Sin embargo,
  - Los circuitos modernos pueden llegar a tener más de  $10^8$  puertas lógicas!
  - El consumo dinámico es proporcional a la frecuencia de conmutación
- El consumo es un problema importante:
  - La energía consumida se transforma en calor, que hay que disipar. Si el circuito consume mucho, puede ser difícil disipar el calor
  - En dispositivos portátiles, el tamaño y el peso de la batería es limitado

# Tecnología CMOS

---

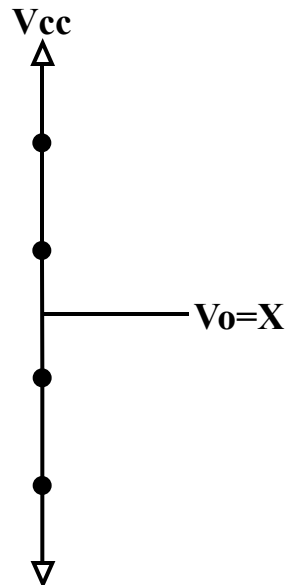
- La tecnología CMOS (Complementary Metal Oxide Semiconductor) es la tecnología más utilizada en la actualidad
- Basada en:
  - Transistores MOS: interruptores controlados por tensión
  - Complementarios: cada transistor o interruptor tiene su complementario, de manera que si un interruptor está abierto su complementario está cerrado y viceversa

# Inversor CMOS

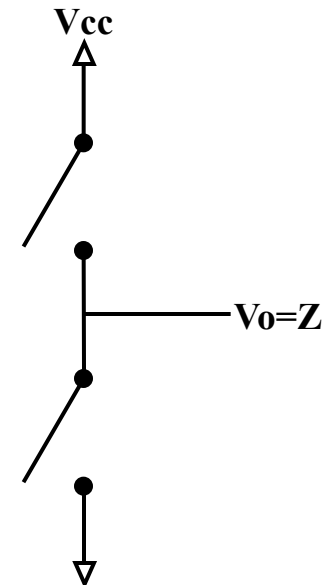


# Valores metalógicos

- Hay situaciones que no se corresponden con valores lógicos
  - Cortocircuito (X)

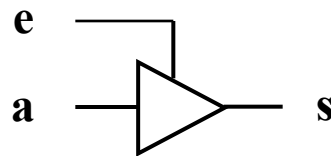


Alta impedancia o triestado (Z)



# Buffer triestado

- Un tipo especial de puerta lógica que puede poner su salida en alta impedancia

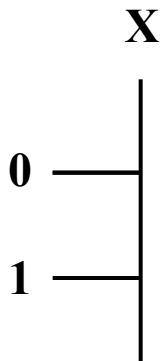


e	a	s
0	0	Z
0	1	Z
1	0	0
1	1	1

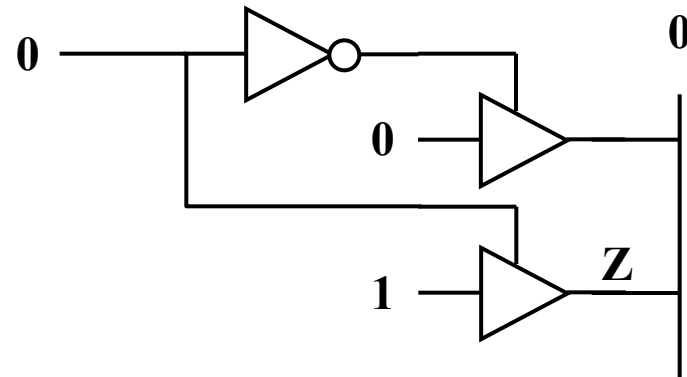


# Buffer triestado

- Los buffers triestado son útiles para permitir múltiples conexiones a un mismo punto evitando cortocircuitos



*Cortocircuito!*



# Familias lógicas

---

- El conjunto de componentes digitales básicos, tales como puertas lógicas y otros que estudiaremos a lo largo del curso, se conoce popularmente como *Serie o Familia 74*
- Existen numerosas subfamilias:
  - Según el rango de temperaturas de operación:
    - Serie 74: 0° a 70°
    - Serie 54: -55° a 125°
  - Según la tecnología utilizada:
    - LS
    - ALS
    - F
    - HC
    - AHC
    - G
    - ....

# Familias lógicas

---

- Designación de componentes:
  - <Serie><Subfamilia><Componente>
- Ejemplo: 74HC00
  - Serie 74: rango de temperaturas convencional
  - Subfamilia HC (High speed CMOS)
  - Componente 00: 4 puertas NAND de 2 entradas
- Importante: las subfamilias no son compatibles entre sí
  - No se deben mezclar componentes de distintas subfamilias en un circuito

# ASICs

---

- Los ASICs son circuitos integrados de aplicación específica
- Una vez diseñado un circuito cualquiera, se puede obtener su “layout” que consiste en un conjunto de máscaras necesarias para la fabricación del circuito

# ASICs

---

- Tamaño: pequeño
- Velocidad: muy alta
  - Tarjeta:  $f < 100$  MHz
  - FPGA: 500 MHz
  - ASIC:  $f < 3$  GHz
- Coste: depende del número de unidades que se fabriquen
  - Coste inicial: diseño y prototipado (100.000€)
  - Coste por pieza: 1-200€
  - Rentables para grandes tiradas (>10.000 piezas/año)
- Fiabilidad: alta; más inmunes al ruido
- Consumo: bajo

# Dispositivos programables

---

- Circuitos programables (PLD, Programmable Logic Devices)
  - Simples
    - PROM: Programmable Read Only Memory
    - PLA: Programmable Logic Array
    - PAL: Programmable Array Logic
    - GAL: Generic Array Logic
  - Complejos
    - CPLD: Complex Programmable Logic Device
    - FPGA: Field Programmable Gate Array

# Tecnologías de circuitos programables

---

- Transistor MOS de puerta flotante (EPROM-FLASH)
  - Transistores que, al aplicarles sobretensión, pueden mantener su tensión de puerta (conexiones programables)
- Memoria RAM estática (SRAM)
  - La memoria permite implementar funciones lógicas
  - Se usan LUTs (Look-Up Tables) de 4 o 5 entradas
- Antifusibles
  - Al fundirse un antifusible se produce un cortocircuito
  - Los cortocircuitos tienen menor resistencia que los diodos-fusibles, proporcionando mayores prestaciones

# Circuitos programables simples

## PLDs (Programmable Logic Devices)

**Entradas + Inversores**

**Matriz  
AND**

**Matriz  
OR**

**Biestables (opcional)**

**Inversores + Salidas**



# Circuitos programables complejos

---

- CPLD:  
Complex Programmable  
Logic Devices
- FPGA:  
Field Programmable  
Gate Array
- Diferencias con los PLDs  
simples
  - Arquitectura
  - Cantidad de recursos  
lógicos
- Fabricantes de  
CPLDs/FPGAs
  - Xilinx
  - Altera
  - Actel
  - Atmel
  - Lattice
  - Cypress

# CPLD: resumen de características

---

- Estructura de PAL con registros y lógica de interconexión
- Capacidad media (hasta 25000 puertas)
- Velocidad media/alta
- Consumo alto
- Tecnología EPROM (reprogramable, no volátil)
- Precio bajo
- La matriz de interconexión global limita el tamaño
- ISP (In-System Programming). JTAG.

# FPGAs

---

- Field Programmable Gate Arrays (Matrices de puertas programables en campo)
- Superan las limitaciones en tamaño de las CPLDs, mediante arquitecturas avanzadas
- Ofrecen mayor variedad de recursos lógicos
  - Lógica combinatorial
  - Lógica secuencial
  - Memoria RAM
  - Conformadores de reloj
  - Señales globales
  - Multiplicadores

## Fabricantes

- Xilinx
- Altera
- Actel
- Atmel

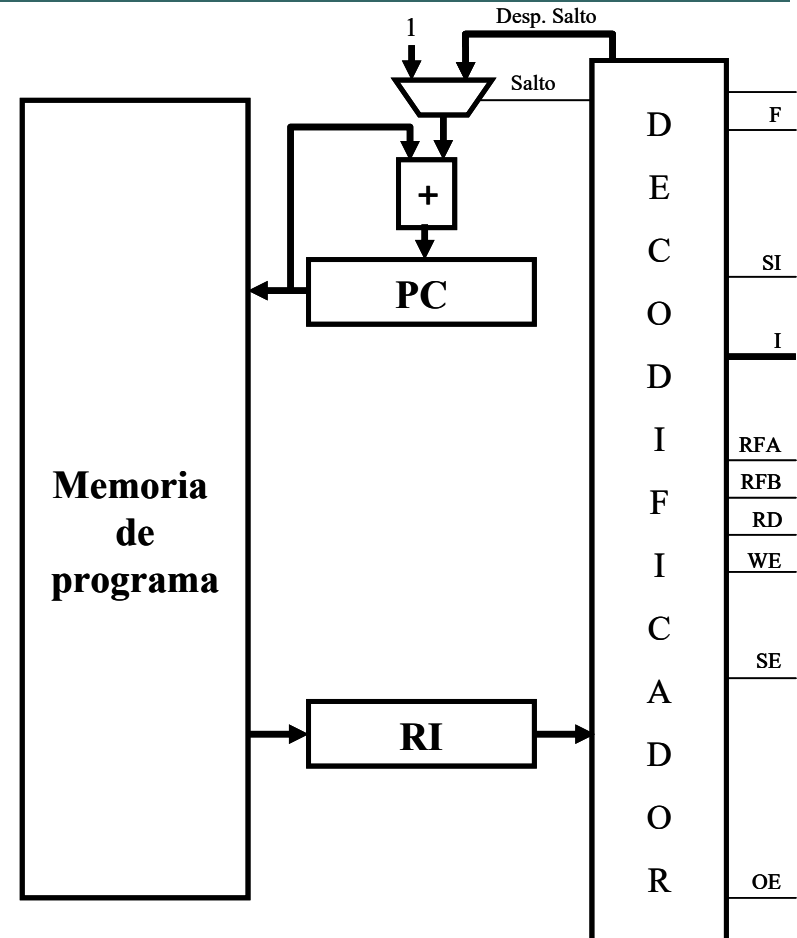
# Microprocesadores

---

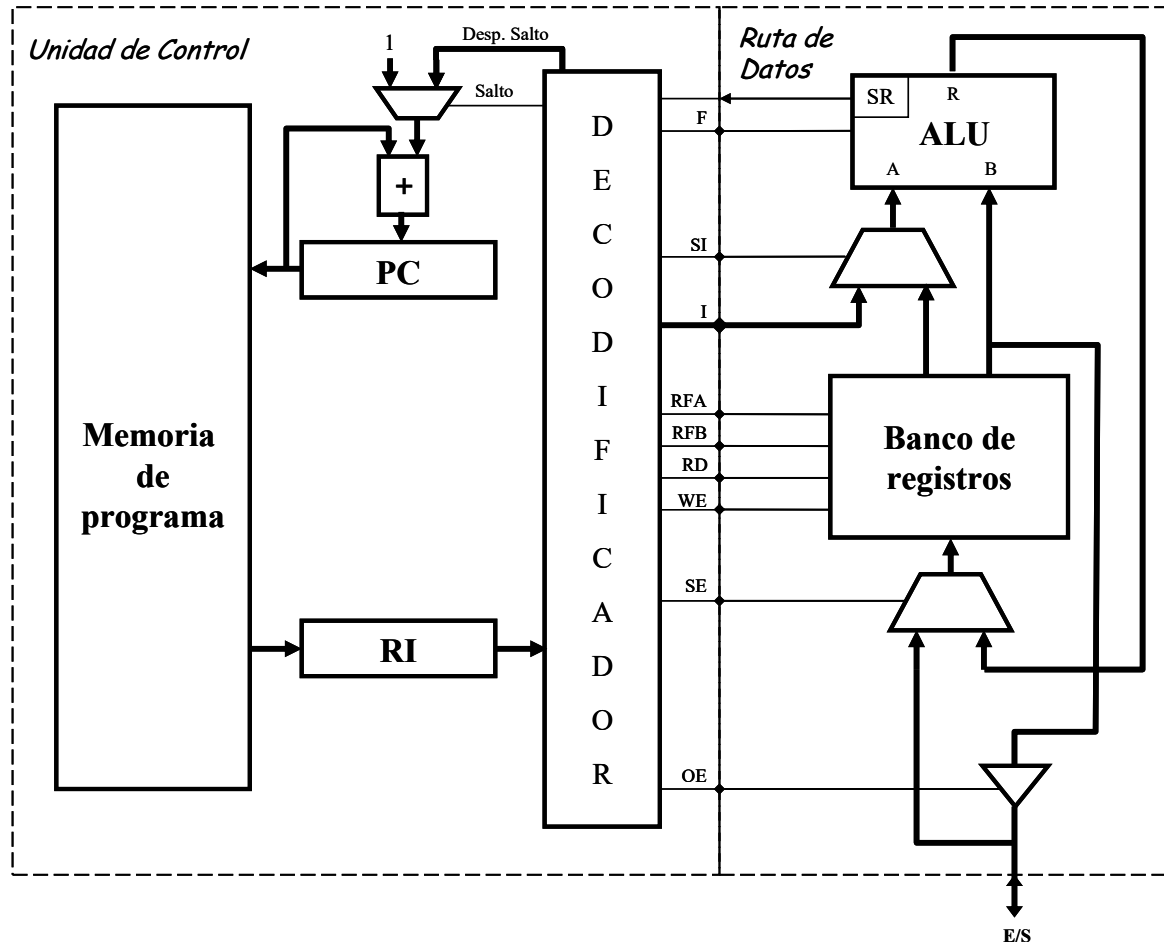
- Microprocesador
  - Orientado a realizar una amplia variedad de algoritmos
- Componentes:
  - Unidad de Proceso o Unidad Central de Proceso (CPU), que se encarga de realizar las operaciones necesarias
    - Ruta de datos general
    - Unidad de control programable (para poder realizar múltiples algoritmos)
  - Memoria, para el almacenamiento de información
  - Unidades de Entrada/Salida, para comunicarse con el exterior

# Unidad de control programable

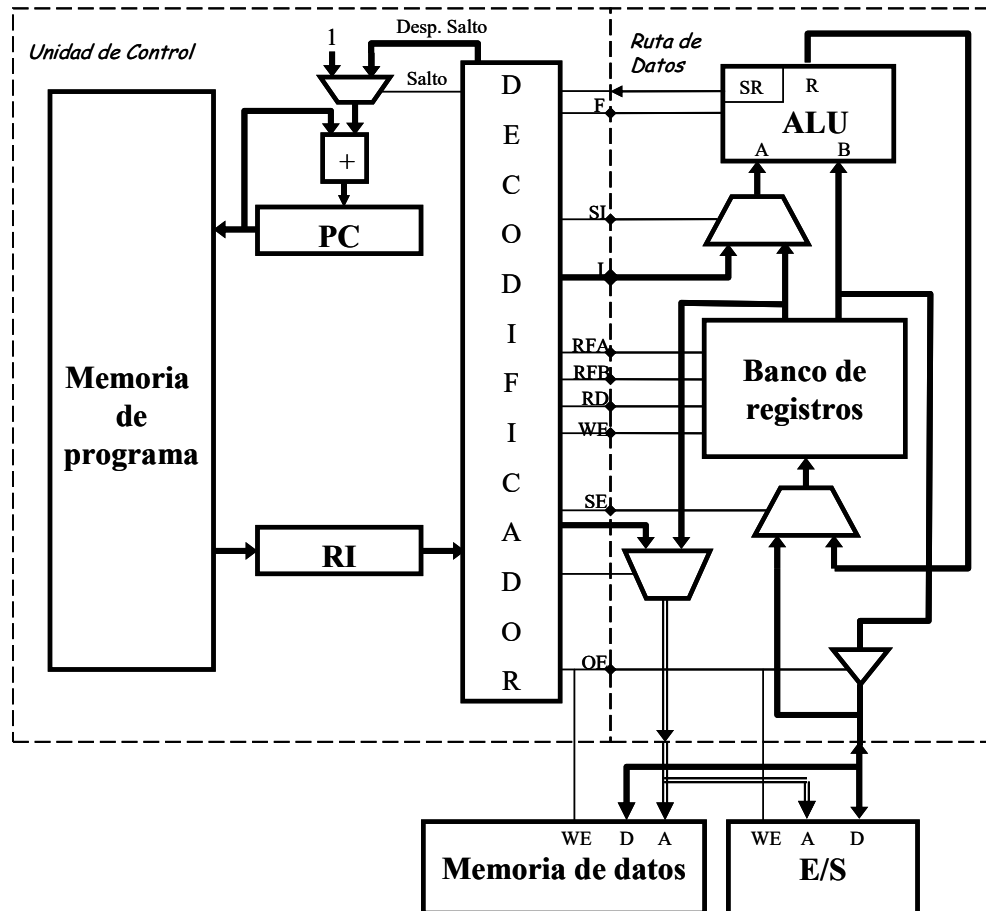
- Las palabras de control se almacenan en una memoria
  - Para ahorrar espacio, se codifican las palabras de control
  - Las palabras de control codificadas las llamamos *instrucciones*
- En cada ciclo de reloj se lee a una instrucción (RI) y se decodifica
- La instrucción correspondiente a cada paso se determina con un contador (PC).
  - Los saltos en la secuencia se realizan con instrucciones, que pueden ser condicionados dinámicamente por SR



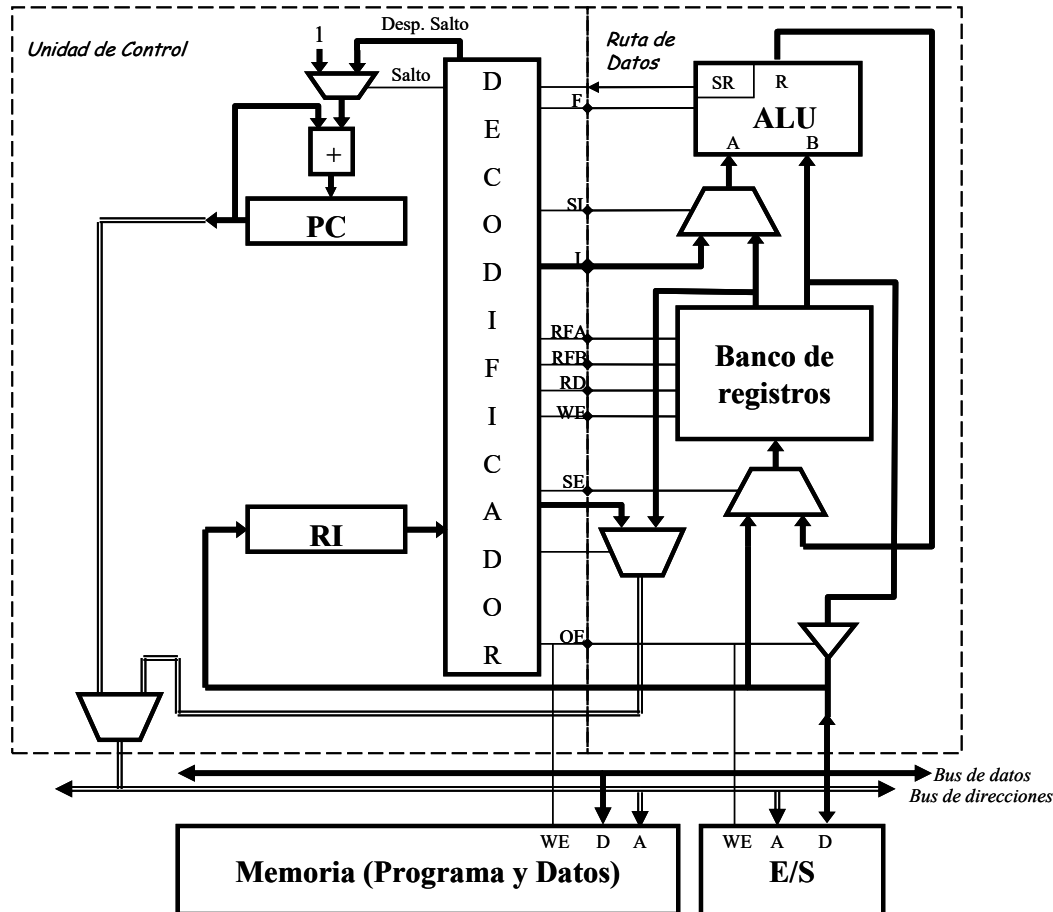
# Microprocesador



# Computador básico (Arquitectura Harvard)



# Computador básico (Arquitectura von Neuman)





# Funcionamiento del computador elemental. Instrucciones

---

- ¿Cómo se ejecutan las instrucciones?
  - Ciclo de instrucción
- ¿Qué tipos de instrucciones hay?
- ¿Cómo se codifica una instrucción?
  - Formato de instrucción
  - Modos de direccionamiento

# Ciclo de instrucción

---

- Cada instrucción se procesa típicamente en 2 fases:
  - Búsqueda de la instrucción: carga la instrucción en el IR
  - Ejecución de la instrucción: decodifica la instrucción y configura la ruta de datos para realizar la operación
- Cada una de estas fases puede realizarse en 1 ciclo de reloj o en varios, dependiendo de la complejidad del microprocesador

# Tipos de instrucciones

---

- Instrucciones de transferencia de datos
  - Transferencia de datos entre registros, entre un registro y la memoria, o entre un registro y un interfaz de E/S
- Instrucciones aritmético-lógicas
  - Realizan operaciones con la ALU: sumas, restas, desplazamientos, etc.
- Instrucciones de salto y bifurcación
  - Permiten realizar cambios en la secuencia de ejecución de las instrucciones
  - Modifican el PC
  - Ejemplos: saltos, llamadas a subrutina, etc.

# Formato de instrucción

---

- Organización de la información de las instrucciones. Las instrucciones se codifican habitualmente por campos:
  - Código de operación (Opcode): indica la operación que se debe realizar
  - Operandos: indican los datos sobre los que se debe operar
- El número de operandos y el tamaño de los campos puede ser variable
- El tamaño de las instrucciones debe ser un múltiplo del ancho de palabra de la memoria

# Modos de direccionamiento

---

- Los operandos pueden indicarse de diversas formas, conocidas como modos de direccionamiento
- Algunos modos usuales:
  - *Inmediato*: el valor del operando se indica en la instrucción
  - *Directo por registro*: la instrucción indica un registro que contiene el operando
  - *Directo a memoria*: la instrucción indica una posición de memoria para el operando
  - *Indirecto*: la instrucción indica un registro que contiene la posición de memoria para el operando

# Ejemplo de formato de instrucción

---

- Para la arquitectura del ejemplo anterior, con los siguientes parámetros:
  - Código de operación: 5 bits
  - Banco de 8 registros (direccionamiento directo con 3 bits)
- Formato de instrucción

Opcode	OpA	OpB	OpD	Otros
--------	-----	-----	-----	-------

# Ejemplos de instrucciones

Operación	Nemónico	Código de operación
Carga dato de memoria	LD	00000
Almacena dato en memoria	ST	00001
Suma	ADD	01000
Resta	SUB	01001
NOT	NOT	01100
AND	AND	01101
OR	OR	01110
Desplazamiento	SHL, SHR	01111
Salto incondicional	JMP	10000
Llamada a subrutina	CALL	10100
Retorno de subrutina	RET	10101
...	...	...

# Ejemplos de instrucciones

- Instrucción de una palabra

Opcode	RFA	RFB	RD	Otros
01000	001	010	011	00

**$R3 = R1 + R2$**

- Instrucción de dos palabras

Opcode	RFA	RFB	RD	Otros
00000	000	000	011	00
Direccion				
1010 1011 1100 1101				

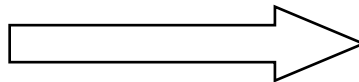
**Carga en R3 el dato  
contenido en la posición de  
memoria ABCDh**



# Lenguaje ensamblador

- Los códigos de instrucción resultan muy poco manejables
- *Lenguaje ensamblador*: las instrucciones se especifican mediante nemónicos y los operandos mediante nombres simbólicos
- Programa ensamblador: traducen las instrucciones de un programa a su código correspondiente
- Ejemplo:

**ADD R1, R2, R3**



0100000101001100

# Bibliografía

---

- Webs de fabricantes:
  - Xilinx: [www.xilinx.com](http://www.xilinx.com)
  - Altera: [www.altera.com](http://www.altera.com)
  - Actel: [www.actel.com](http://www.actel.com)
  - Lattice: [www.latticesemi.com](http://www.latticesemi.com)
- “Principios de Diseño Digital”. D. Gajski. Ed. Prentice Hall
- “Fundamentos de Sistemas Digitales”. Thomas L. Floyd. Pearson Prentice Hall
- “Sistemas digitales: principios y aplicaciones”, Tocci, Ronald J. Pearson Prentice Hall
- “Dispositivos lógicos programables (PLD): diseño práctico de aplicaciones”. García Iglesias, José Manuel. RaMa