

Sistemas embebidos II

Sesión 2

Programación

Las FPGA programan utilizando lenguajes de descripción de hardware (HDL), que son diferentes de los lenguajes de programación tradicionales porque describen el comportamiento del hardware en lugar de instrucciones secuenciales para una CPU.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Declaración de la entidad para la compuerta AND
entity AndGate is
    Port (
        A : in STD_LOGIC; -- Entrada A
        B : in STD_LOGIC; -- Entrada B
        Y : out STD_LOGIC -- Salida Y
    );
end AndGate;

-- Descripción de la arquitectura del comportamiento
architecture Behavioral of AndGate is
begin
    -- Proceso concurrente que asigna la salida Y como el resultado de A AND B
    Y <= A and B;
end Behavioral;
```

```
def and_gate(a, b):
    """
    Función que simula una compuerta AND de dos entradas.

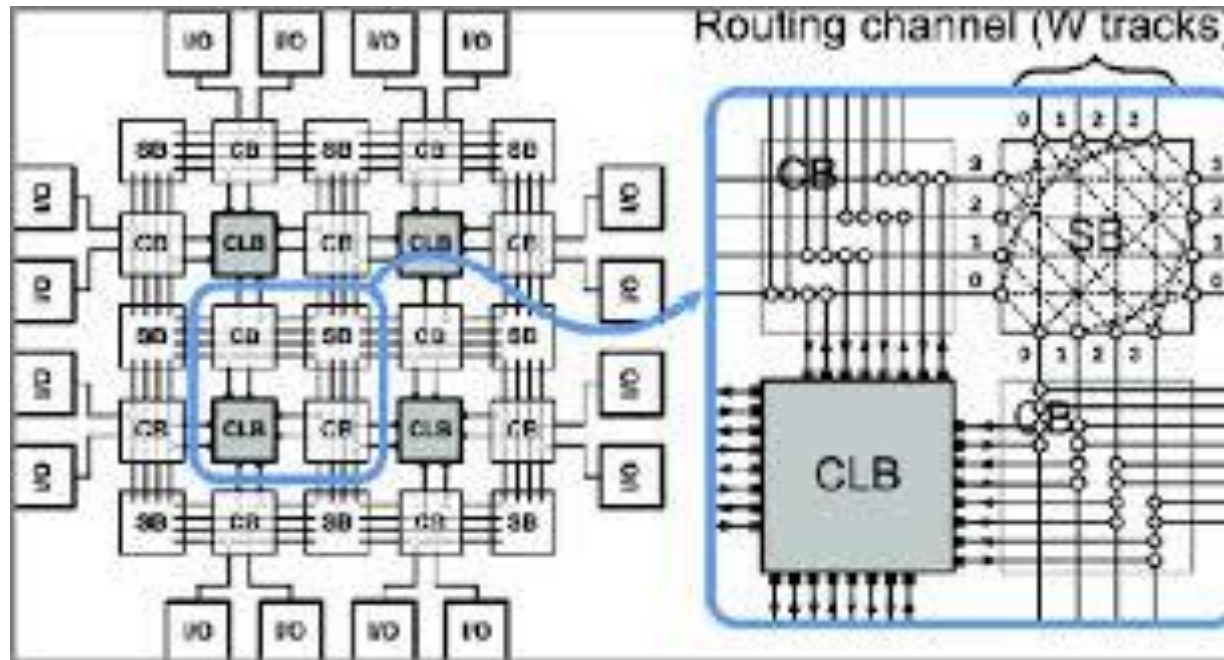
    Args:
        a (int): Primera entrada (0 o 1)
        b (int): Segunda entrada (0 o 1)

    Returns:
        int: Salida de la compuerta AND (0 o 1)
    """
    return a & b # Operador bitwise AND en Python

# Ejemplo de uso
entrada_a = 1 # Simulando una entrada alta (1)
entrada_b = 0 # Simulando una entrada baja (0)
salida_y = and_gate(entrada_a, entrada_b)
print(f"La salida de la compuerta AND es ↓ salida_y")
```

Programación (VHDL - HDL)

HDL es un término genérico que abarca todos los lenguajes de descripción de hardware. Un HDL permite describir el comportamiento de un circuito digital de manera que puede ser sintetizado para implementarse en hardware como FPGAs, ASICs, etc.

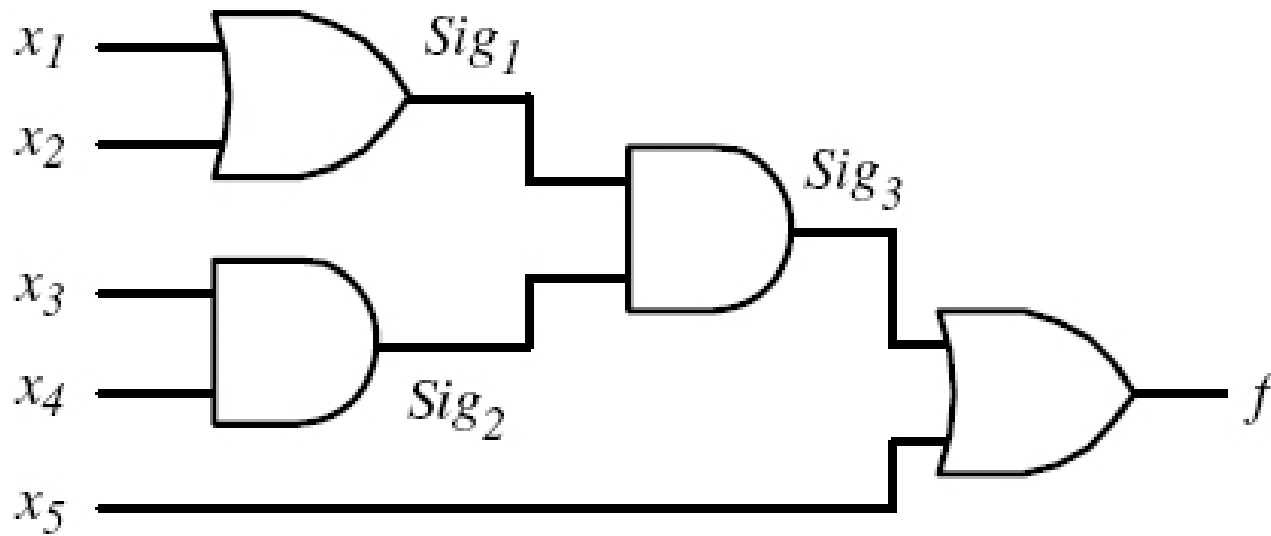


Programación (VHDL - HDL)

- VHDL: El Departamento de Defensa de EE. UU. Desarrolló VHDL, un lenguaje VHSIC (circuitos integrados de muy alta velocidad). El Departamento investigó el idioma durante la década de 1980 para crear circuitos de alta velocidad. Fue adoptado como estándar IEEE 1076 en el año 1987.
- Verilog HDL: Warmke, C.-L, creó el lenguaje. Huang, P. Goel y P. Moorby para realizar una simulación y modelar puertas en 1984 para un simulador lógico. Cadence se hizo cargo del proyecto en 1990 y adquirió el estándar IEEE 1364 en 1995.

```
case stCur is
  when stReceive =>
    lee <= '0';
    escribe <= '0';
    if (pide_cuenta='1' and RD='1') then
      case Entrada is
        when "00110000" => Salida<=Latchx(7 downto 0);
        when "00110001" => Salida<=Latchx(15 downto 8);
        when "00110010" => Salida<=Latchx(23 downto 16);
        when "00110011" => Salida<=Latchx(31 downto 24);
        when "00110100" => Salida<=Latchy(7 downto 0);
        when "00110101" => Salida<=Latchy(15 downto 8);
        when "00110110" => Salida<=Latchy(23 downto 16);
        when "00110111" => Salida<=Latchy(31 downto 24);
        when "00111000" => Salida<=Latchz(7 downto 0);
        when "00111001" => Salida<=Latchz(15 downto 8);
        when "00111010" => Salida<=Latchz(23 downto 16);
        when "00111011" => Salida<=Latchz(31 downto 24);
        when others => Salida<="00110000";
      end case;
      stNext <= stSend;
    else
      stNext <= stReceive;
    end if;
  when stSend =>
    lee <= '1';
    escribe <= '1';
    stNext <= stReceive;
end case;
```

Programación (VHDL - HDL)



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Declaración de la entidad para el circuito lógico
entity LogicCircuit is
    Port (
        x1 : in STD_LOGIC; -- Entrada x1
        x2 : in STD_LOGIC; -- Entrada x2
        x3 : in STD_LOGIC; -- Entrada x3
        x4 : in STD_LOGIC; -- Entrada x4
        x5 : in STD_LOGIC; -- Entrada x5
        f  : out STD_LOGIC -- Salida f
    );
end LogicCircuit;

-- Descripción de la arquitectura del comportamiento
architecture Behavioral of LogicCircuit is
    signal Sig1, Sig2, Sig3 : STD_LOGIC; -- Señales internas
begin
    -- Operación AND entre x1 y x2
    Sig1 <= x1 and x2;

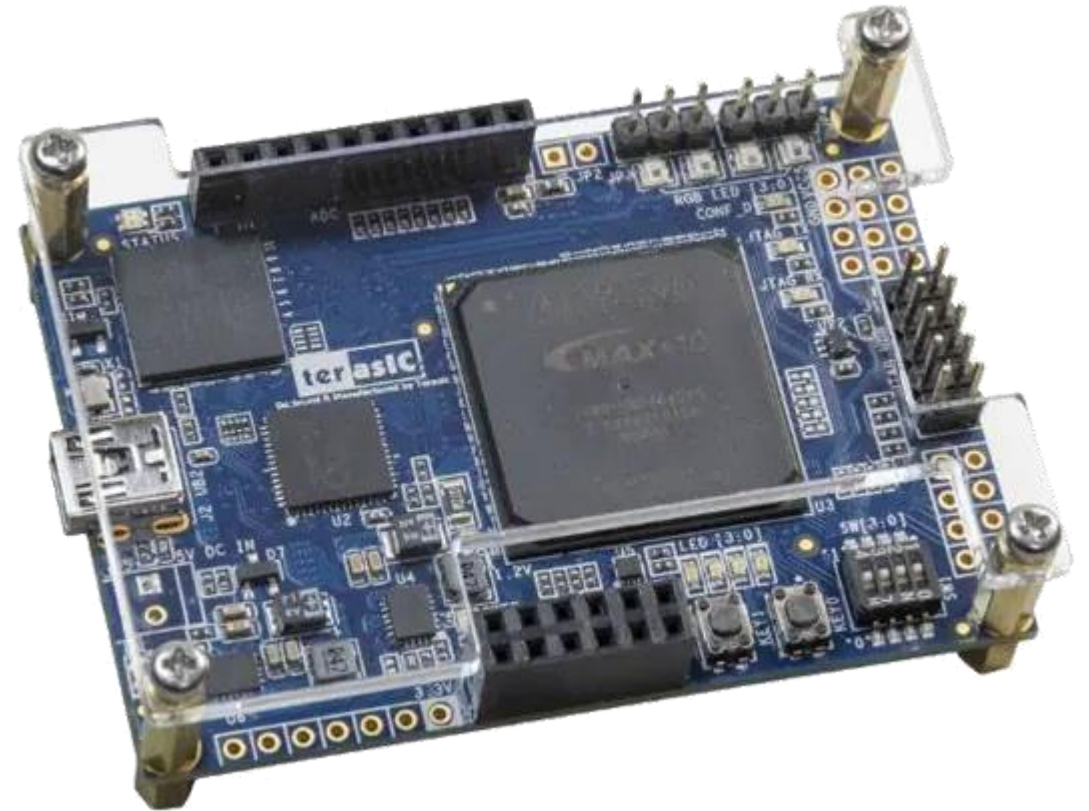
    -- Operación AND entre x3 y x4
    Sig2 <= x3 and x4;

    -- Operación AND entre Sig1 y Sig2
    Sig3 <= Sig1 and Sig2;

    -- Operación OR entre Sig3 y x5
    f <= Sig3 or x5;
end Behavioral;
```

FPGA

Es un tipo de dispositivo semiconductor que se puede programar para realizar funciones lógicas específicas. A diferencia de los microprocesadores tradicionales, que siguen un conjunto predefinido de instrucciones, las FPGA son matrices de puertas lógicas programables que permiten a los usuarios configurar y reconfigurar el hardware según sus necesidades específicas.



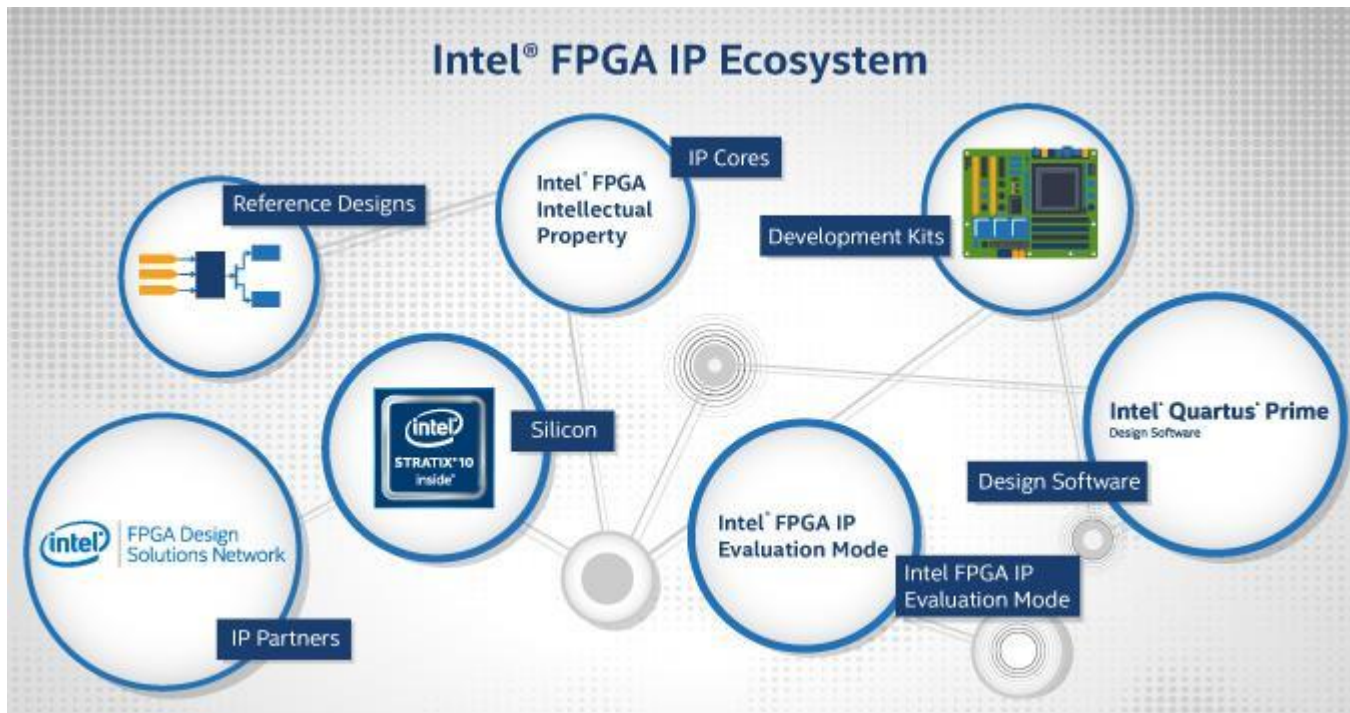
FPGA

- **Programabilidad:** Las FPGA son "field-programmable", lo que significa que pueden ser configuradas después de su fabricación. Esto permite que los diseñadores personalicen la funcionalidad del hardware para adaptarse a aplicaciones específicas.
- **Reconfigurabilidad:** Las FPGA pueden ser reprogramadas cuantas veces sea necesario. Esta capacidad es particularmente útil durante las fases de desarrollo y prueba de productos, ya que permite hacer ajustes o cambios sin necesidad de fabricar nuevos chips.
- **Paralelismo:** Las FPGA pueden ejecutar múltiples operaciones simultáneamente, en contraste con los microprocesadores, que generalmente ejecutan operaciones de forma secuencial. Esta capacidad es ideal para aplicaciones que requieren procesamiento intensivo, como el procesamiento de señales, la compresión de video y audio, y algoritmos criptográficos.
- **Consumo de energía ajustable:** A diferencia de los ASIC, que tienen un consumo de energía fijo una vez fabricados, las FPGA permiten a los diseñadores optimizar el consumo de energía según las necesidades específicas de la aplicación.

FPGA

- **Alto rendimiento:** Las FPGA pueden ofrecer un rendimiento muy alto, especialmente para aplicaciones que requieren cálculos complejos y que pueden ser paralelizados. Su capacidad de realizar procesamiento en tiempo real las hace ideales para aplicaciones críticas como la computación de alto rendimiento y sistemas de control industrial.
- **Integración de bloques de IP:** Muchas FPGA modernas vienen con bloques de IP predefinidos, como microcontroladores, bloques de DSP, interfaces de comunicación y más, que pueden ser utilizados para acelerar el desarrollo de sistemas complejos.

Bloque IP (Núcleo de propiedad intelectual de semiconductores)



Es un plano de diseño de una unidad de lógica reutilizable, celda o circuito integrado (comúnmente llamado "chip") que es propiedad intelectual de una de las partes. Los núcleos PI pueden ser licenciados a otra de las partes o pueden ser poseído y usados por una sola parte. El término deriva del licenciamiento de las patentes y/o derechos de autor que existen en el diseño. Los núcleos IP pueden ser usados como bloques de construcción en diseños de circuitos integrados de aplicación específica (ASIC) o de FPGA.

FPGA

- **Procesamiento de señales digitales (DSP):** Para aplicaciones como el procesamiento de imágenes, video, y señales de radar.
- **Redes y telecomunicaciones:** Implementación de protocolos de comunicación, enrutamiento de alta velocidad y codificación/decodificación de señales.
- **Sistemas embebidos:** Control de dispositivos en tiempo real, robótica, automotriz y aeroespacial.
- **Criptografía y seguridad:** Implementación de algoritmos criptográficos y funciones de seguridad avanzadas.
- **Prototipado y desarrollo de ASICs:** Uso como plataforma de prototipado para diseñar y probar ASICs antes de la fabricación.

DSP

Es un sistema basado en un procesador o microprocesador que posee un conjunto de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad. Debido a esto es especialmente útil para el procesamiento de señales y representación de señales analógicas en tiempo real: en un sistema que trabaje de esta forma (tiempo real) se reciben muestras normalmente provenientes de un conversor analógico/digital (ADC).



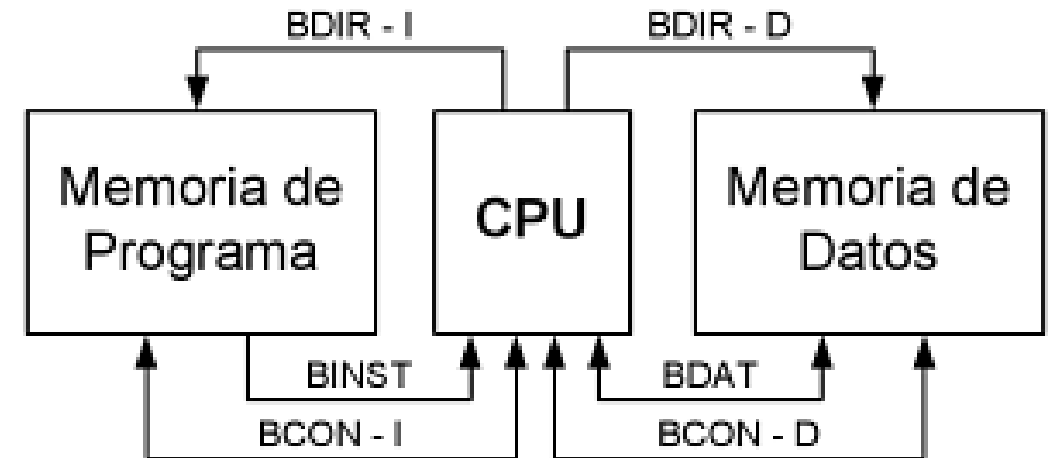
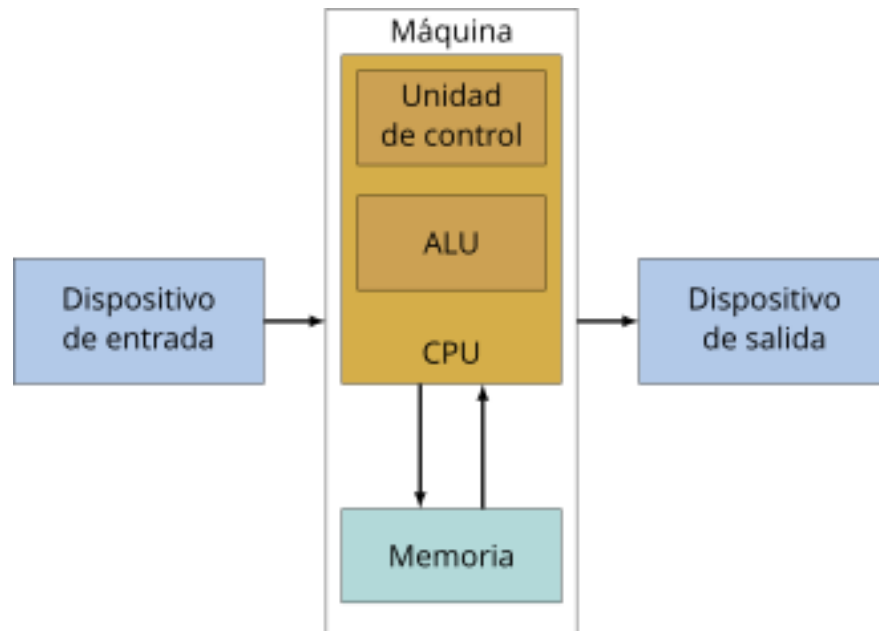
DSP

- Conversores en las entradas y salidas
- Memoria de datos, memoria de programa y DMA.
- MACs: multiplicadores y acumuladores.
- ALU: unidad aritmético-lógica.
- Registros.
- PLL: bucles enganchados en fase.
- PWM: módulos de control de ancho de pulso.



Von Neumann Vs Harvard

Los DSP abandonan la arquitectura clásica de Von Neumann, en la que datos y programas están en la misma zona de memoria, y apuestan por la denominada Arquitectura Harvard. En una arquitectura Harvard existen bloques de memoria físicamente separados para datos y programas. Cada uno de estos bloques de memoria se direcciona mediante buses separados (tanto de direcciones como de datos), e incluso es posible que la memoria de datos tenga distinta anchura de palabra que la memoria de programa (como ocurre en ciertos microcontroladores).



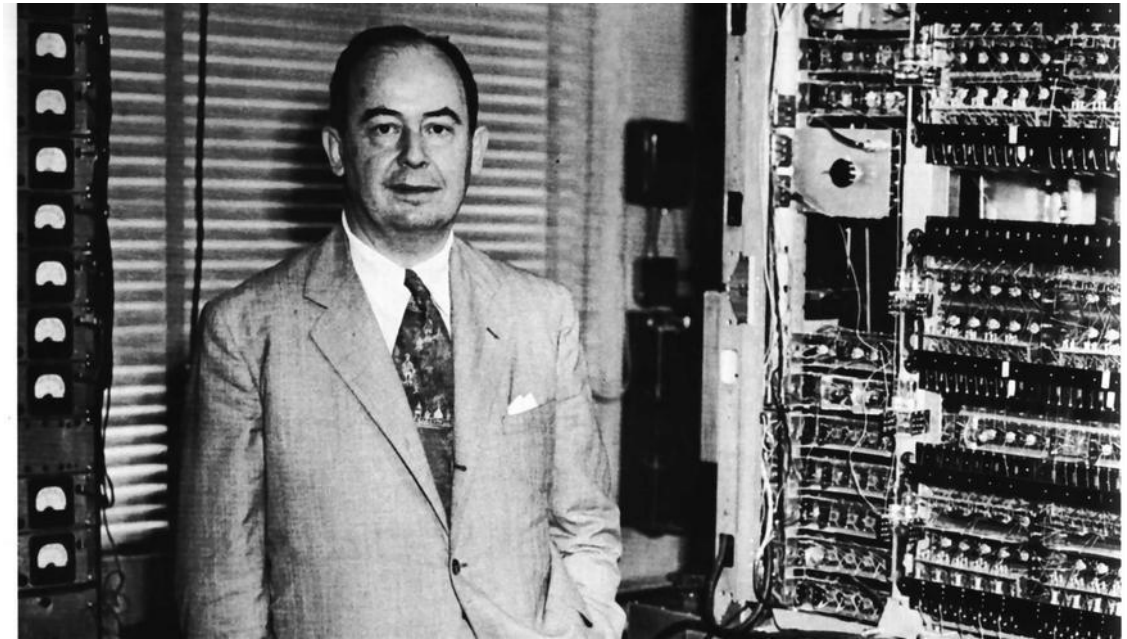
Von Neumann Vs Harvard

La arquitectura de Von Neumann es un diseño de computadora que utiliza una sola memoria compartida para almacenar tanto instrucciones como datos.

- **Memoria Unificada:** Una única memoria almacena tanto los datos como las instrucciones del programa.
- **Búsqueda Secuencial:** Las instrucciones y los datos se buscan secuencialmente desde la misma memoria, lo que puede llevar a un "cuello de botella" de rendimiento conocido como "von Neumann bottleneck".
- **Flexibilidad:** Debido a su diseño unificado, es más fácil de programar y es más flexible para modificaciones de software.

Ejemplos

- Computadoras Personales y Servidores
- Microcontroladores Simples
- Supercomputadoras Antiguas



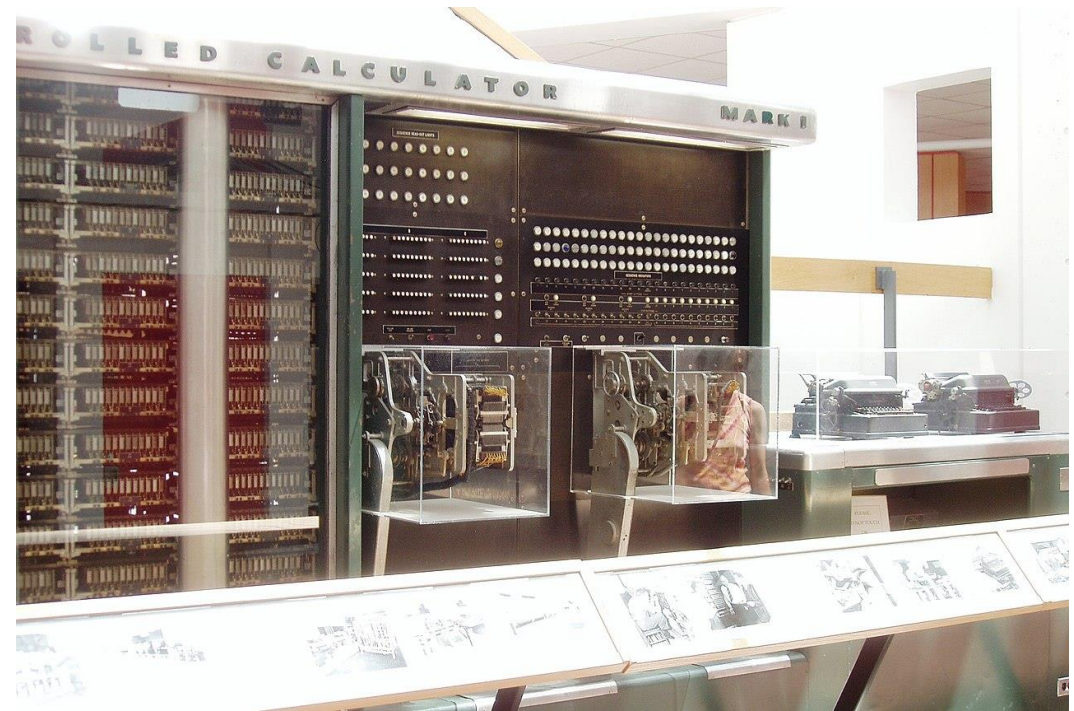
Von Neumann Vs Harvard

La arquitectura Harvard es un diseño de computadora que separa físicamente las memorias para almacenar instrucciones (código) y datos.

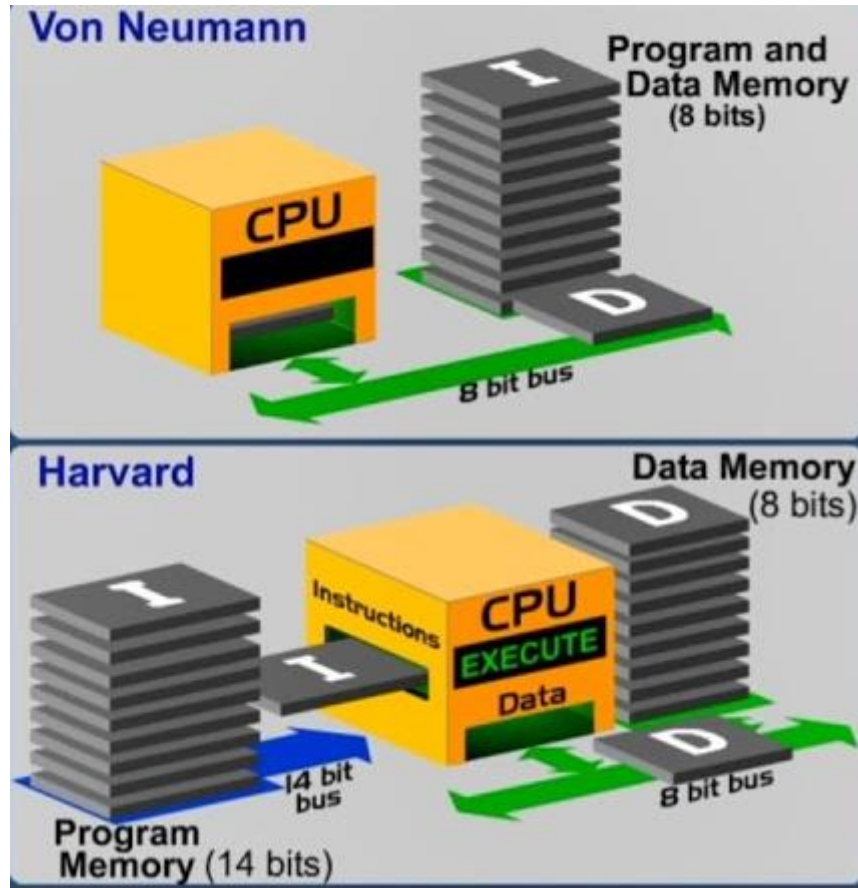
- **Memoria Separada:** Tiene memorias independientes para instrucciones y datos, lo que permite que las instrucciones y los datos se busquen simultáneamente.
- **Mayor Ancho de Banda:** Al tener buses separados para instrucciones y datos, se puede lograr un mayor rendimiento ya que el procesador puede leer instrucciones y datos al mismo tiempo sin esperar.
- **Seguridad:** Ofrece mayor seguridad y control sobre el acceso a la memoria, ya que los datos no pueden ser tratados como instrucciones (y viceversa).

Ejemplos

- Microcontroladores y DSPs
- Microcontroladores AVR (ATmega de Atmel).
- Procesadores RISC (Reduced Instruction Set Computer)



Von Neumann Vs Harvard



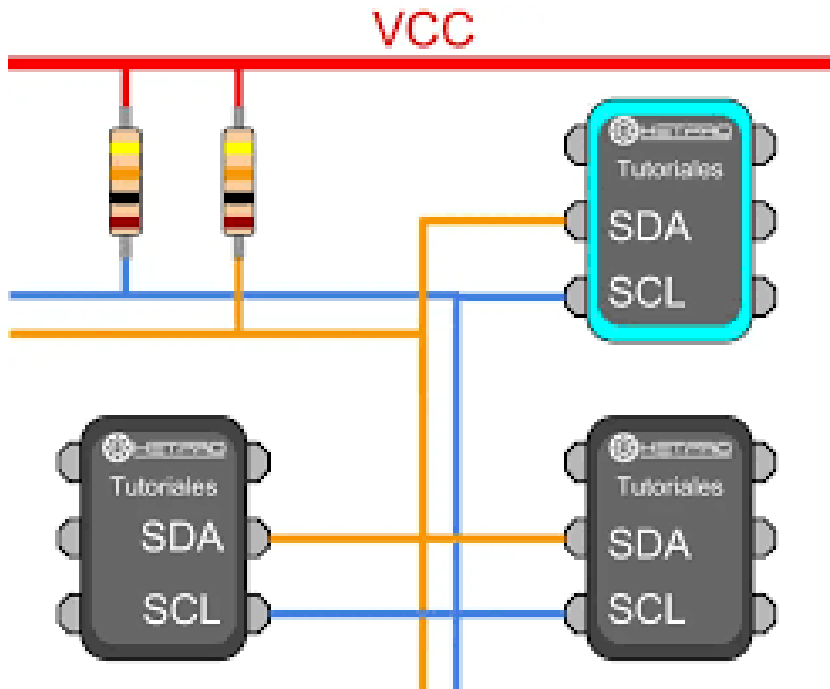
- Arquitectura de von Neumann es más flexible y sencilla de programar, lo que la hace ideal para aplicaciones generales en computadoras personales, servidores y algunos microcontroladores de propósito general.
- Arquitectura Harvard ofrece un mayor rendimiento en aplicaciones donde la velocidad de acceso a la memoria es crítica, como en microcontroladores para sistemas embebidos y procesadores de señales digitales (DSPs).

DSP

- **Arquitectura de Hardware Especializada:** Los DSPs suelen tener una arquitectura de hardware que optimiza la ejecución de operaciones matemáticas comunes en el procesamiento de señales, como la multiplicación-acumulación (MAC) en una sola instrucción de ciclo.
- **Alta Velocidad de Procesamiento:** Están diseñados para ejecutar operaciones matemáticas, especialmente operaciones de punto flotante y punto fijo, a alta velocidad
- **Capacidad de Manejo de Datos en Tiempo Real:** Los DSPs están optimizados para manejar flujos de datos en tiempo real, lo que significa que pueden procesar datos a medida que se reciben sin necesidad de almacenamiento intermedio extenso.
- **Memoria de Acceso Rápido y Jerarquizada:** Incorporan memorias de acceso rápido, como registros, cachés de instrucciones y datos, y bancos de memoria interna para reducir la latencia de acceso a los datos.



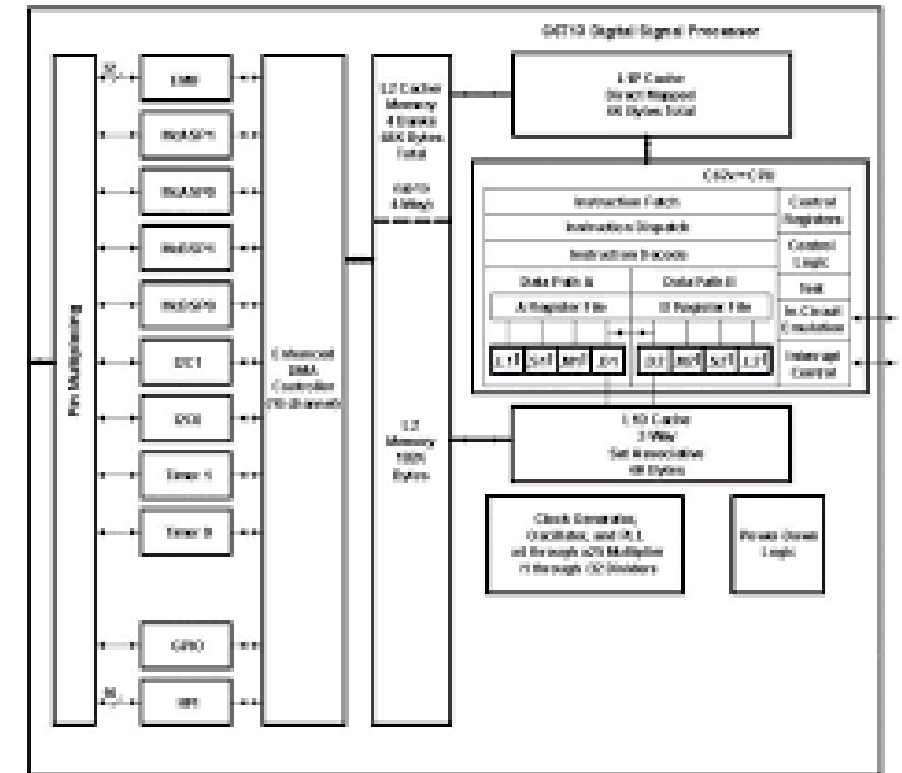
DSP



- **Bajo Consumo de Energía:** Los DSPs están diseñados para maximizar la eficiencia energética. Esto es particularmente importante en dispositivos portátiles o embebidos, donde el consumo de energía es crítico.
- **Instrucciones de Procesamiento de Señal Dedicadas:** Tienen un conjunto de instrucciones extendido que incluye operaciones específicas para el procesamiento de señales, como instrucciones de multiplicación-acumulación, transformadas rápidas (FFT), filtrado, y otras funciones matemáticas complejas.
- **Capacidad de Interfaz con Periféricos de Alta Velocidad:** Los DSPs a menudo deben interactuar con una variedad de periféricos de alta velocidad, como convertidores analógico-digital (ADC) y digital-analógico (DAC), módulos de comunicación y dispositivos de almacenamiento. Para esto, suelen incluir interfaces de E/S (Entrada/Salida) de alta velocidad, como SPI, I2C, y UART.

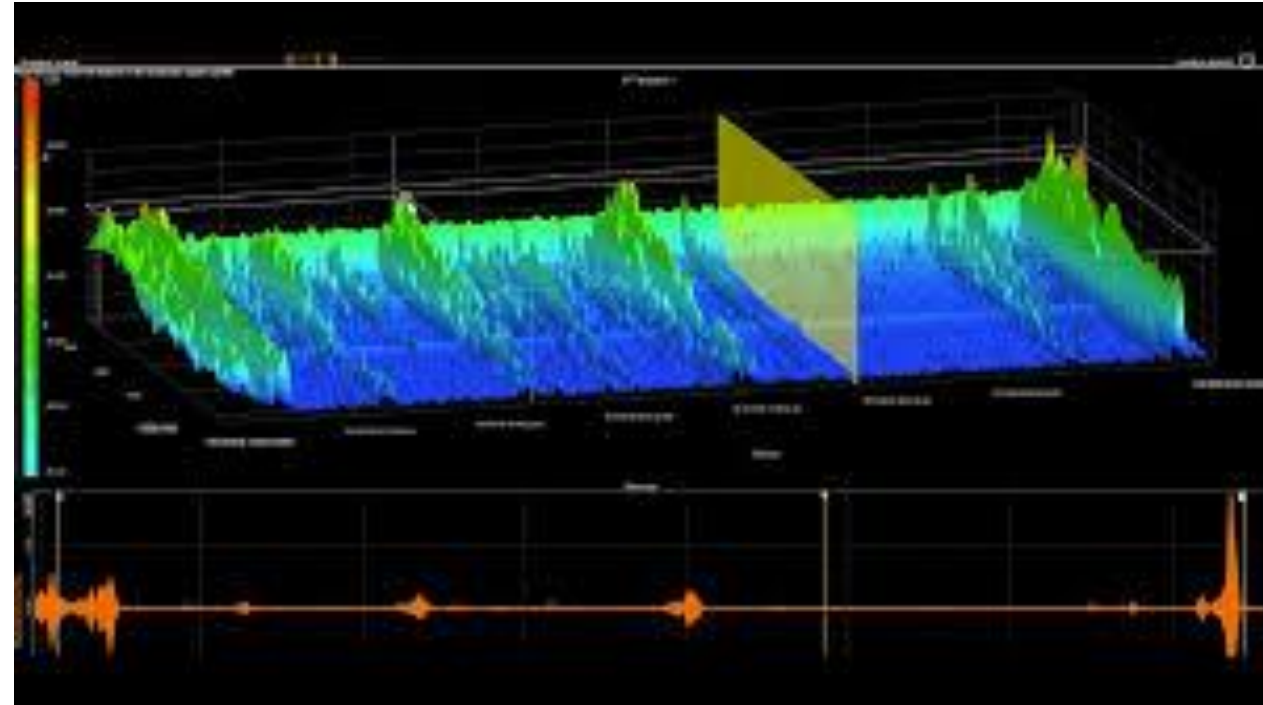
DSP

- **Pipeline Profundo y Múltiples Etapas de Ejecución:** Tienen pipelines profundos que permiten que múltiples etapas de una instrucción se ejecuten simultáneamente. Esta característica mejora significativamente la velocidad de procesamiento, permitiendo una mayor eficiencia y rendimiento
- **Capacidad de Bajo Retardo (Low Latency):** Los DSPs están diseñados para operar con bajo retardo, lo cual es esencial para aplicaciones en tiempo real donde cada milisegundo cuenta, como en la eliminación de eco, procesamiento de señales en comunicaciones y procesamiento de audio.



DSP

- Procesamiento de Audio y Voz
 - Codificación
 - Decodificación
 - Eliminación de ruido
 - Ecualización
- Procesamiento de Imágenes y Video
 - Compresión de video
 - Reconocimiento de patrones
 - Estabilización de imagen.
- Telecomunicaciones
 - Modulación y demodulación de señales
 - Corrección de errores gestión de protocolos de red
- Control Automático y Robótica
 - Sistemas de control en tiempo real
 - Procesamiento de sensores
 - Visión por computadora.



DSP

algunos DSP admiten lenguajes de alto nivel como LabVIEW y Matlab, facilitando la programación mediante compilación y mejorando la eficiencia en el desarrollo de algoritmos complejos. Comúnmente son programados en C

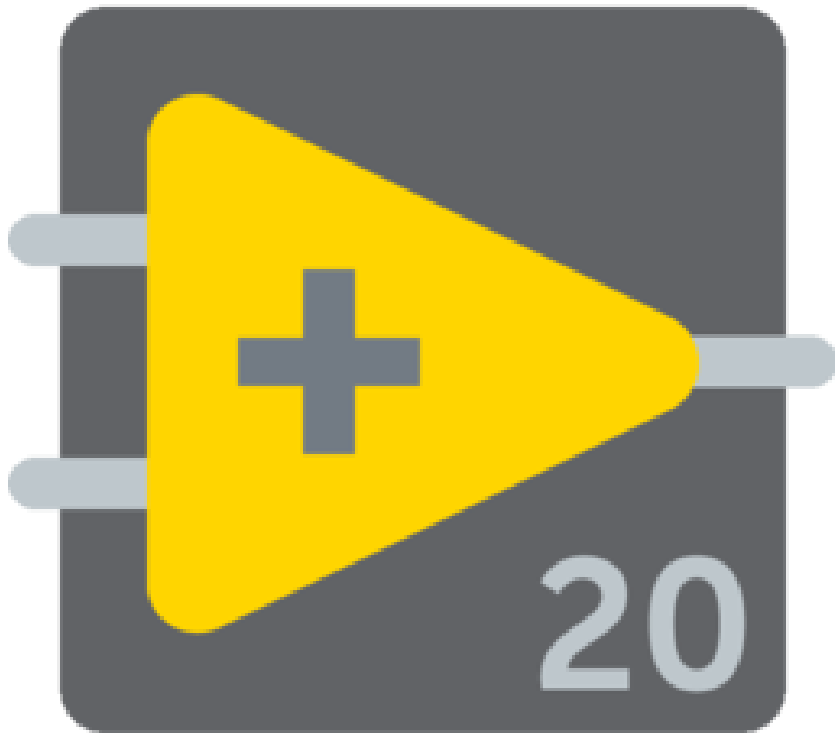
https://books.google.com.co/books?id=9ef6Tkw8YwEC&printsec=frontcover&source=gbg_ge_summary_r&cad=0#v=onepage&q&f=false

<https://www.ni.com/docs/en-US/bundle/flexrio/page/digital-signal-processing-dsp-overview.html?srsId=AfmBOooQ9buwFr1J3D2Vw3AGuxKYO4MumUCpEZ4wxyw7xzkaAiwi-W-A>

DSP



DSP



¡GRACIAS!



Esta licencia permite a otros distribuir, remezclar, retocar, y crear a partir de esta obra de manera no comercial y, a pesar que sus nuevas obras deben siempre mencionar a la IU Digital y mantenerse sin fines comerciales, no están obligados a licenciar obras derivadas bajo las mismas condiciones.



IUDigital
de Antioquia
INSTITUCIÓN UNIVERSITARIA DIGITAL DE ANTIOQUIA