

Sistemas embebidos II

Sesión 1

Presentación

Juan Pablo Restrepo Uribe

Ingeniero Biomédico

Magister en Automatización y Control Industrial

juan.restrepo@iudigital.edu.co

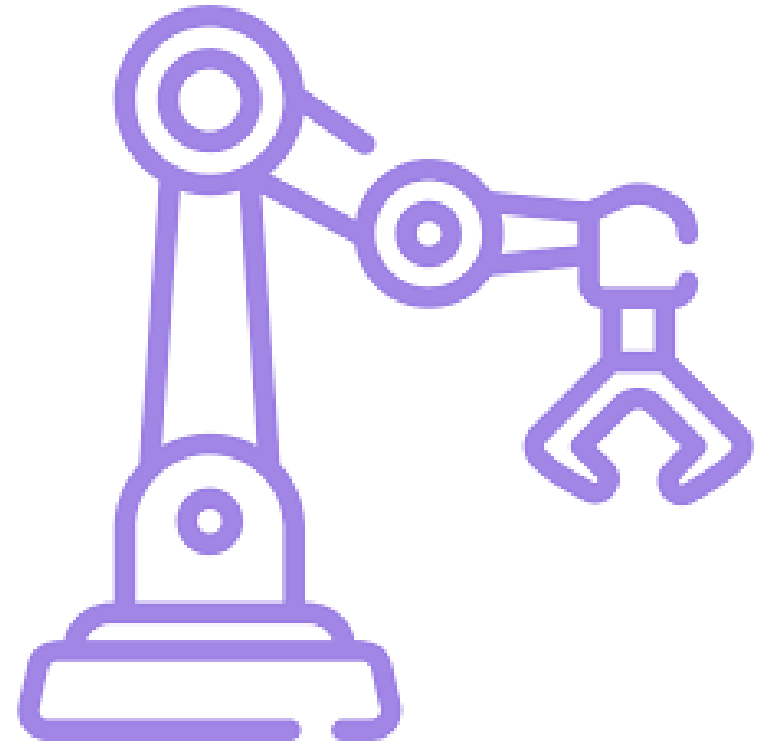
Pregunta orientadora

¿De qué forma podemos aplicar los conceptos relacionados con los sistemas embebidos, y las diferentes técnicas y dispositivos que existen para la implementación de sistemas digitales más complejos, que permitan la solución de problemas específicos dentro de un contexto departamental y nacional?



Objetivo general

Reconocer y analizar las necesidades de diseño, con el fin de resolver problemas mediante la integración efectiva de hardware y software.



Mapa del curso



Sistemas Digitales Embebidos II

UNIDAD 1

Introducción a los sistemas embebidos

- Introducción al hardware de DSP y FPGA
- Tecnología para FPGA
- Elementos FPGA para algoritmos DSP
- Aritmética esencial de un DSP
- Técnicas de gráficos de flujos de señales SFG

UNIDAD 2

Procesamiento de FPGA

- Procesado en ritmo de frecuencia
- Procesado de señales multitasa para FPGA
- Técnicas CORDIC

UNIDAD 3

Aplicaciones de DSP y FPGA





- Algoritmos adaptativos
- Comunicaciones habilitadas por DSP y FPGA
- Problemas de temporización y sincronización

Actividad de conocimientos previos

En la actualidad, los sistemas electrónicos embebidos se ven afectados por dos elementos cruciales: el hardware y los algoritmos o software específicos que son capaces de ejecutar dicho hardware. Es necesario lograr un equilibrio entre ambos componentes, con el fin de garantizar un óptimo funcionamiento de nuestro dispositivo.

1. Teniendo en cuenta lo anterior, queremos conocer tu opinión de cómo, a través de los sistemas embebidos, se pueden generar algoritmos y firmwares eficientes alrededor de sistemas como las FPGA.
2. Es importante que, con los conocimientos que has adquirido hasta el momento: en el curso de Sistemas Digitales Embebidos I y mediante tu experiencia, argumentes tu respuesta.
3. También, te pedimos presentar dos casos prácticos en los que optimices uno y otro elemento del sistema embebido. Si consideras que ninguno de los elementos debería ser optimizado, ¡por supuesto que es válido!
4. Discute tus ideas en el foro, intenta dar respuesta a las preguntas planteadas según tus conocimientos y revisa las respuestas de tus compañeros. ¡Qué sea la oportunidad para conversar y debatir las nuevas ideas!

Tareas

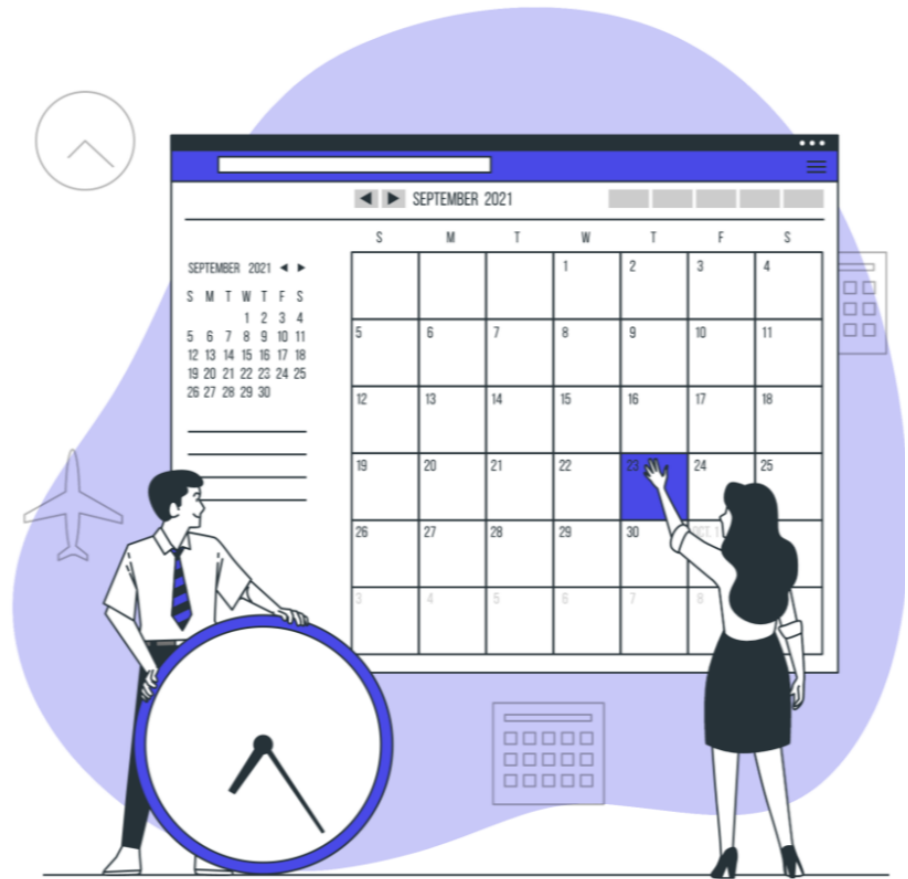
⋮ ▾ Actividad 1		20% del total + ⋮
⋮	 Evidencia de aprendizaje Taller sobre los SP y las FPGA Unidad 1 Módulo Fecha de entrega 1 de sep en 23:59 100 pts	  ⋮
⋮ ▾ Actividad 2		25% del total + ⋮
⋮	 Proyecto integrador Parte 1: Diseño de filtros digitales Unidad 2 Módulo Fecha de entrega 15 de sep en 23:59 100 pts	  ⋮
⋮ ▾ Actividad 3		25% del total + ⋮
⋮	 Evidencia de aprendizaje 3: Proyecto integrador -Parte 2- Aplicaciones de FPGA Unidad 3 Módulo Fecha de entrega 29 de sep en 23:00 100 pts	  ⋮
⋮ ▾ Actividad presencial		30% del total + ⋮
⋮	 Evidencia de aprendizaje 4: Proyecto integrador -Parte 3- Diseño de un sistema (presencial) Unidad 3 Módulo Fecha de entrega 5 de oct en 12:00 100 pts	  ⋮

Tareas

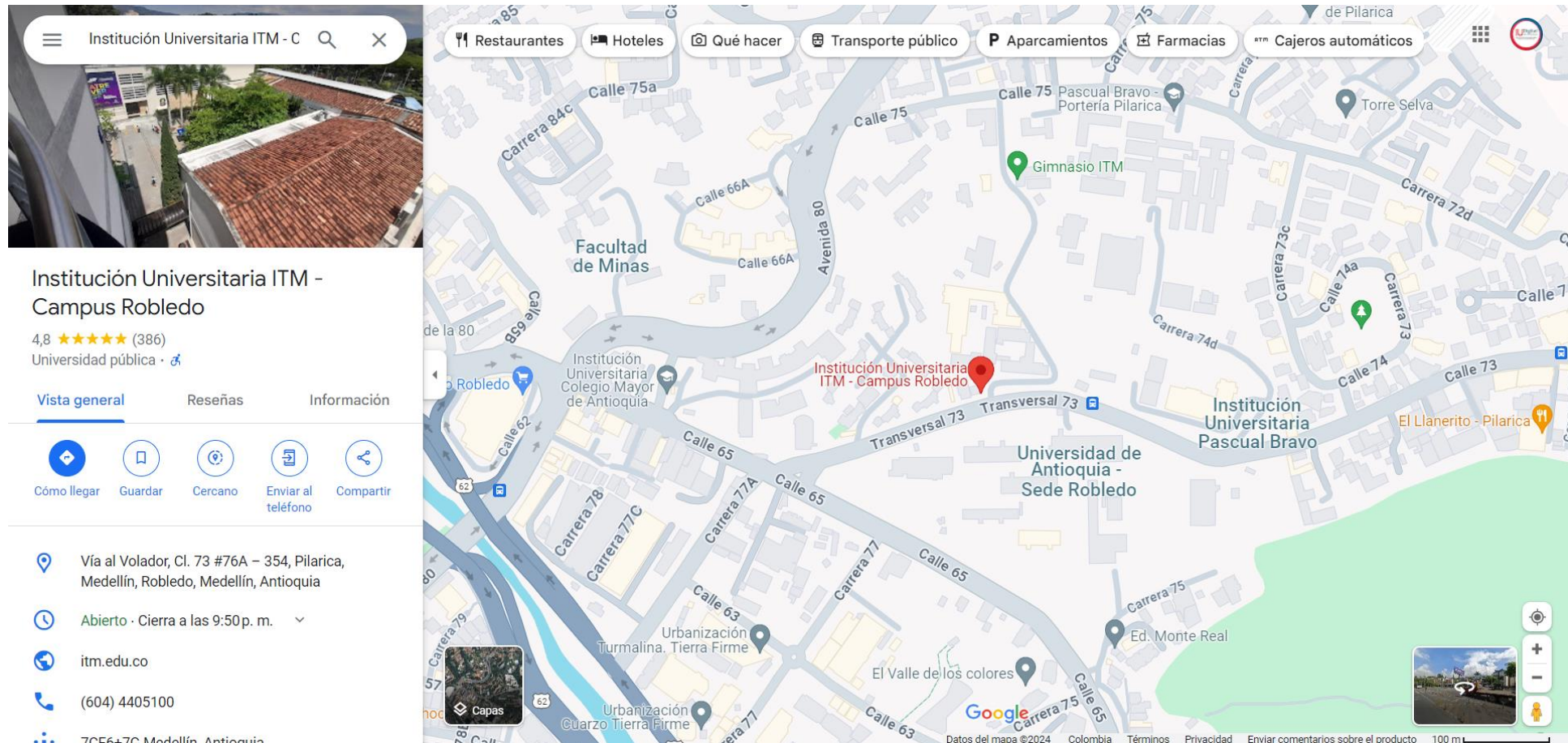
Tareas		0% del total	
	Otros conceptos para el procesamiento de datos Unidad 2 Módulo		
	Actividad de Refuerzo Unidad 3 Módulo 99.99 pts		
	Aplicaciones de algoritmos adaptativos Unidad 3 Módulo		
	Arquitectura de las FPGA's Unidad 1 Módulo		
	Elemento de la arquitectura FPGA: Bloques Lógicos Programables (CLB) Unidad 1 Módulo		

Encuentro presencial

Sábado 5 de octubre de 1 PM A 6 PM
ITM - Robledo



Encuentro presencial



Para tener en cuenta



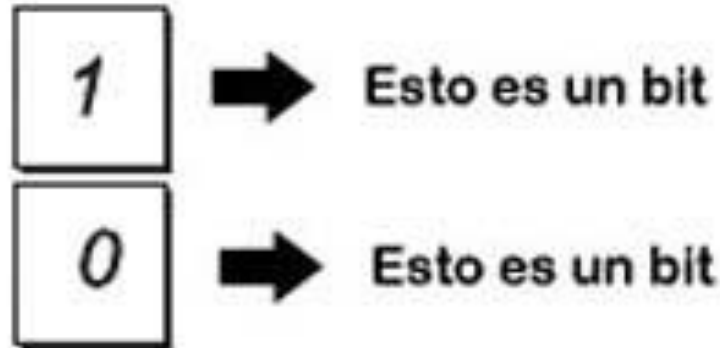
Sistemas de numeración

Sistema	Base	Dígitos	Ejemplo
Binario	2	0 – 1	$1001010_{(2)} \rightarrow 74_{(10)}$
Octal	8	0 – 1 – 2 – 3 – 4 – 5 – 6 – 7	$6523_{(8)} \rightarrow 3411_{(10)}$
Decimal	10	0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9	$9856_{(10)}$
Hexadecimal	16	0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – A – B – C – D – E – F	$59852AE_{(16)} \rightarrow 93868718_{(10)}$

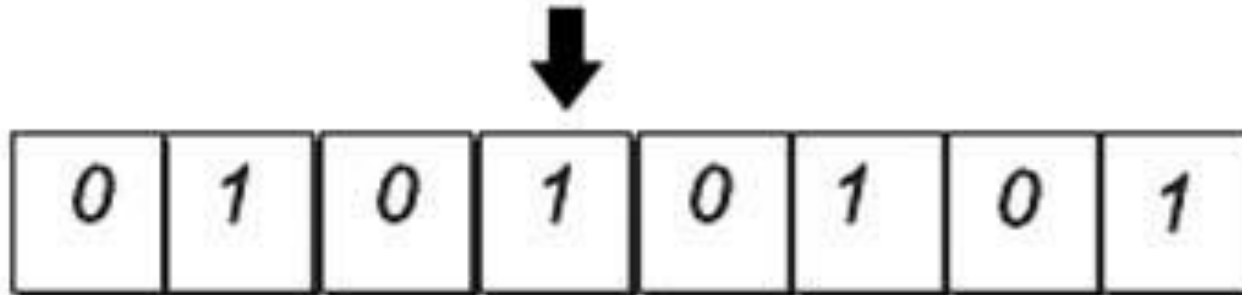
Sistemas de numeración

Número decimal	Número octal	Octal en binario	Número Hexadecimal	Hexadecimal en binario
0	0	0 0 0	0	0 0 0 0
1	1	0 0 1	1	0 0 0 1
2	2	0 1 0	2	0 0 1 0
3	3	0 1 1	3	0 0 1 1
4	4	1 0 0	4	0 1 0 0
5	5	1 0 1	5	0 1 0 1
6	6	1 1 0	6	0 1 1 0
7	7	1 1 1	7	0 1 1 1
8	10	-	8	1 0 0 0
9	11	-	9	1 0 0 1
10	12	-	A	1 0 1 0
11	13	-	B	1 0 1 1
12	14	-	C	1 1 0 0
13	15	-	D	1 1 0 1
14	16	-	E	1 1 1 0
15	17	-	F	1 1 1 1

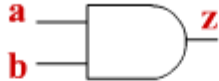





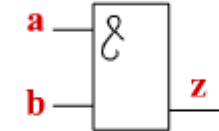
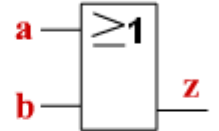
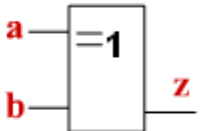
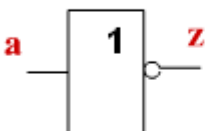
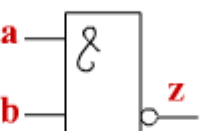
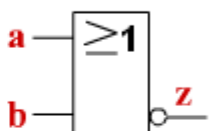
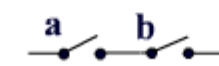

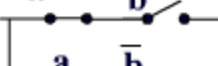

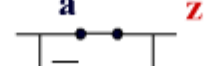
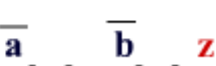
¿Qué es un bit?



Esto es un byte



Lógica binaria

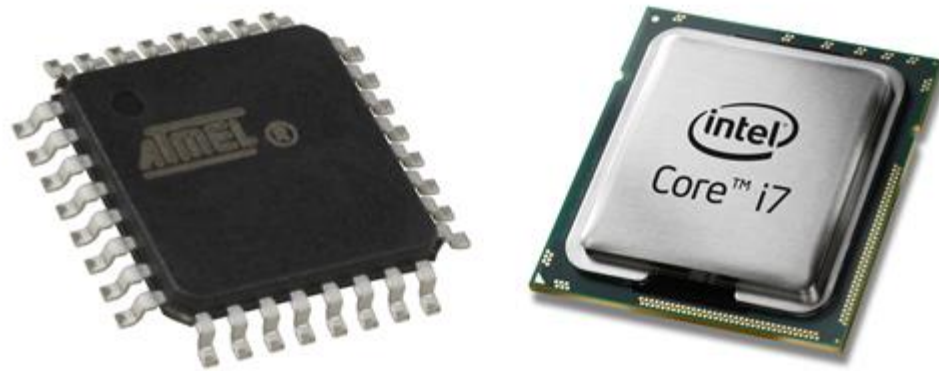
NOMRE	AND - Y	OR - O	XOR O-exclusiva	NOT Inversor	NAND	NOR																																																																																	
SÍMBOLO																																																																																							
SÍMBOLO																																																																																							
TABLA DE VERDAD	<table><tr><th>a</th><th>b</th><th>z</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	a	b	z	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><th>a</th><th>b</th><th>z</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	a	b	z	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><th>a</th><th>b</th><th>z</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	a	b	z	0	0	0	0	1	1	1	0	1	1	1	0	<table><tr><th>a</th><th>z</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	a	z	0	1	1	0	<table><tr><th>a</th><th>b</th><th>z</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	a	b	z	0	0	1	0	1	1	1	0	1	1	1	0	<table><tr><th>a</th><th>b</th><th>z</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	a	b	z	0	0	1	0	1	0	1	0	0	1	1	0
a	b	z																																																																																					
0	0	0																																																																																					
0	1	0																																																																																					
1	0	0																																																																																					
1	1	1																																																																																					
a	b	z																																																																																					
0	0	0																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	1																																																																																					
a	b	z																																																																																					
0	0	0																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	0																																																																																					
a	z																																																																																						
0	1																																																																																						
1	0																																																																																						
a	b	z																																																																																					
0	0	1																																																																																					
0	1	1																																																																																					
1	0	1																																																																																					
1	1	0																																																																																					
a	b	z																																																																																					
0	0	1																																																																																					
0	1	0																																																																																					
1	0	0																																																																																					
1	1	0																																																																																					
EQUIVALENTE EN CONTACTOS																																																																																							
AXIOMA	$z = a \cdot b$	$z = a + b$	$z = \bar{a} \cdot b + a \cdot \bar{b}$	$z = \bar{a}$	$z = \overline{a \cdot b}$	$z = \overline{a + b}$																																																																																	

Lógica binaria

$$S = (A + B) * (\bar{A} + B) * C$$

A	B	C	D	F(A,B,C,D)
1	1	1	1	0
1	1	1	0	1
1	1	0	1	0
1	1	0	0	1
1	0	1	1	0
1	0	1	0	1
1	0	0	1	0
1	0	0	0	1
0	1	1	1	0
0	1	1	0	1
0	1	0	1	0
0	1	0	0	1
0	0	1	1	0
0	0	1	0	1
0	0	0	1	0
0	0	0	0	1

Microcontrolador vs Microprocesador



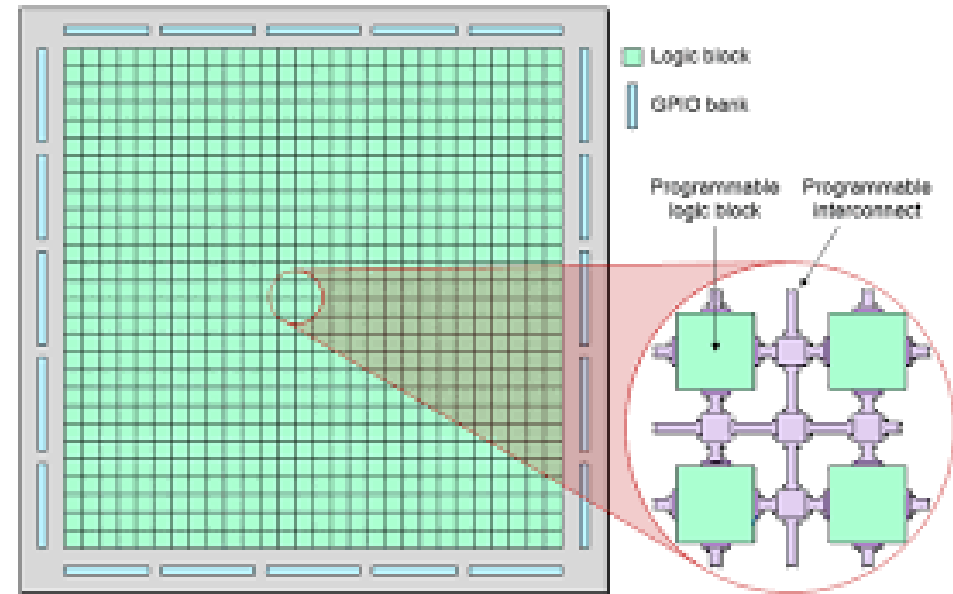
Microcontrolador vs Microprocesador

Son circuitos integrados utilizados en dispositivos electrónicos con diferentes aplicaciones y características:

- **Microcontrolador**
 - Es un sistema completo en un solo chip que incluye un microprocesador, memoria (RAM, ROM/Flash), y periféricos (temporizadores, puertos de entrada/salida, ADC, etc.)
 - Diseñado para realizar tareas específicas en sistemas embebidos.
 - Utilizado en aplicaciones como electrodomésticos, controladores industriales, dispositivos médicos, y sistemas automotrices.
- **Microprocesador:**
 - Es el componente principal de un sistema de procesamiento, que contiene la unidad central de procesamiento (CPU) pero no incluye memoria o periféricos en el mismo chip.
 - Diseñado para realizar una amplia gama de tareas de procesamiento.
 - Utilizado en computadoras personales, servidores, y dispositivos que requieren alta capacidad de cálculo y flexibilidad.

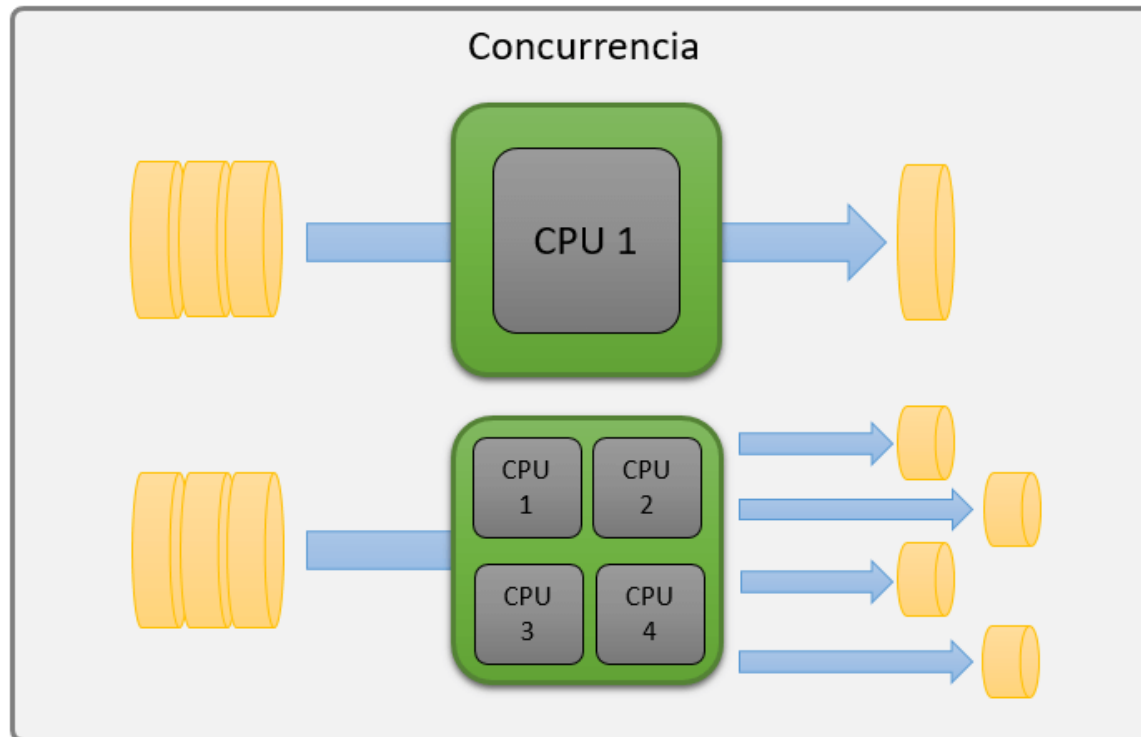
Microcontrolador vs Microprocesador Vs FPGA

Consiste en una matriz de bloques lógicos programables interconectados por una red de interconexión. A diferencia de los microcontroladores y microprocesadores, una FPGA no tiene una arquitectura fija; se puede configurar para realizar diferentes funciones lógicas según las necesidades del usuario.



Tiene una arquitectura configurable a nivel de hardware. El usuario puede diseñar circuitos personalizados que se ejecutan en paralelo, proporcionando una gran flexibilidad y rendimiento específico para ciertas aplicaciones.

Microcontrolador vs Microprocesador Vs FPGA



- **Microcontrolador:** Usado en aplicaciones embebidas y específicas donde se requiere control directo de hardware con bajo consumo de energía.
- **Microprocesador:** Usado en sistemas donde se requiere alta capacidad de procesamiento y flexibilidad, como en computadoras y servidores.
- **FPGA:** Usado en aplicaciones donde se necesita un alto grado de paralelismo, procesamiento en tiempo real, y flexibilidad para implementar hardware personalizado.

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.

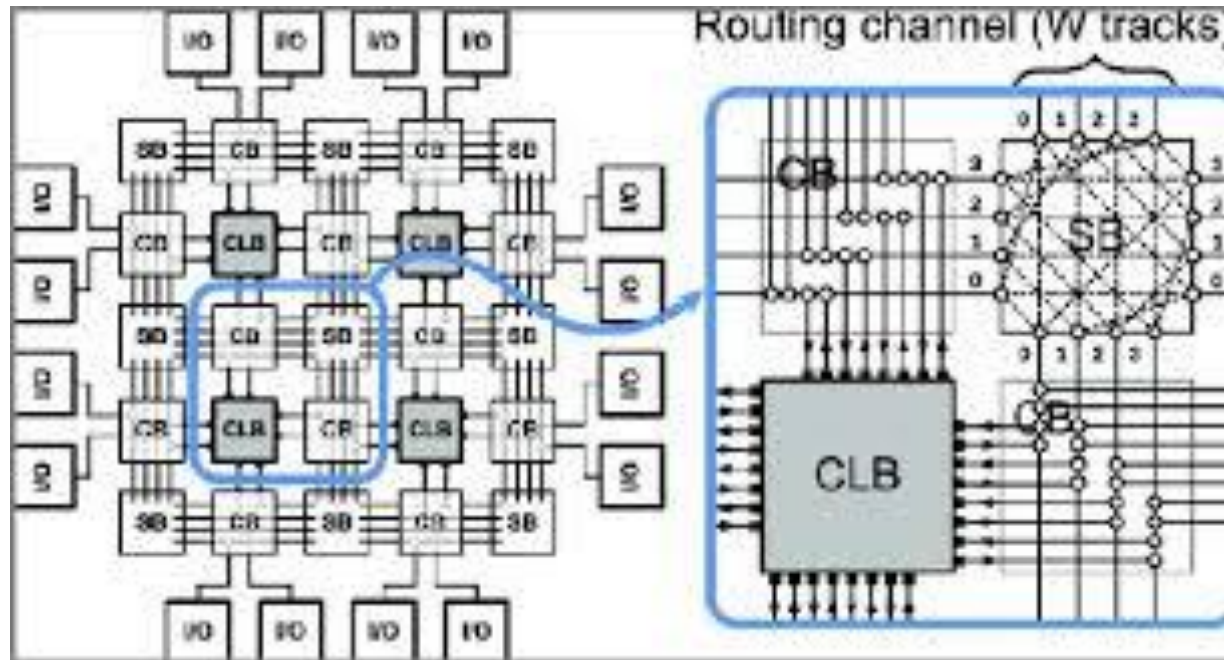
    Args:
        a (int): Primera entrada (0 o 1)
        b (int): Segunda entrada (0 o 1)

    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.

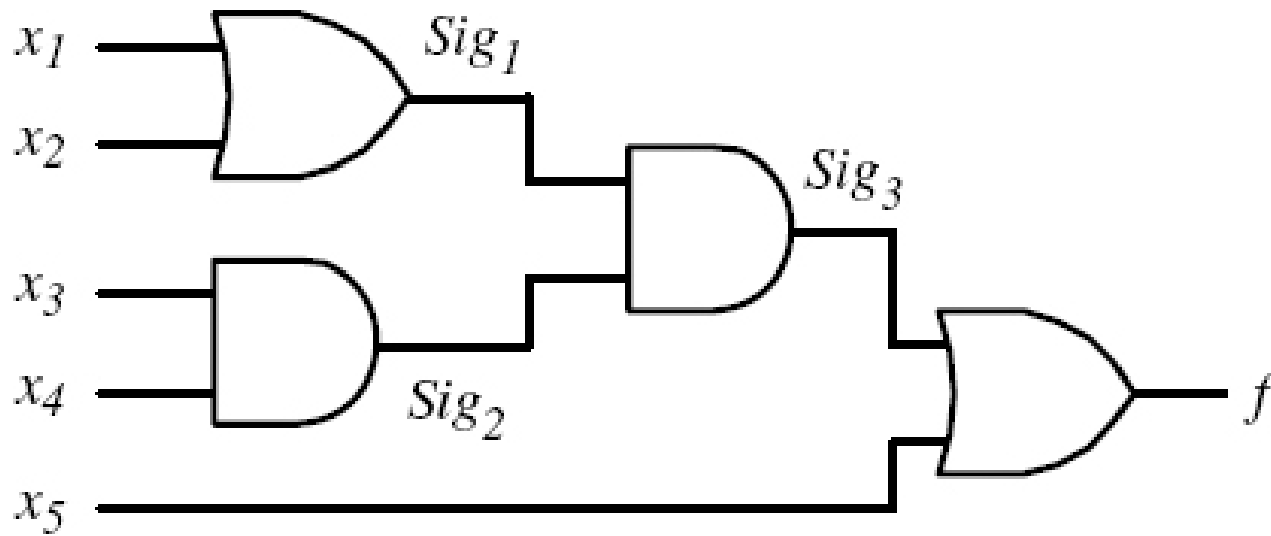


Programación (VHDL - HDL)

- 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;
```


¡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