

♪ **FPGAeduDesign**



INTRODUCCIÓN AL DISEÑO DIGITAL

Placa de desarrollo FPGA NANO 1K

ELEMSYS

Electronics and Embedded Systems

por: Roly Sandro Gutierrez Benito



Li-UNSAAC



BIOTRONK



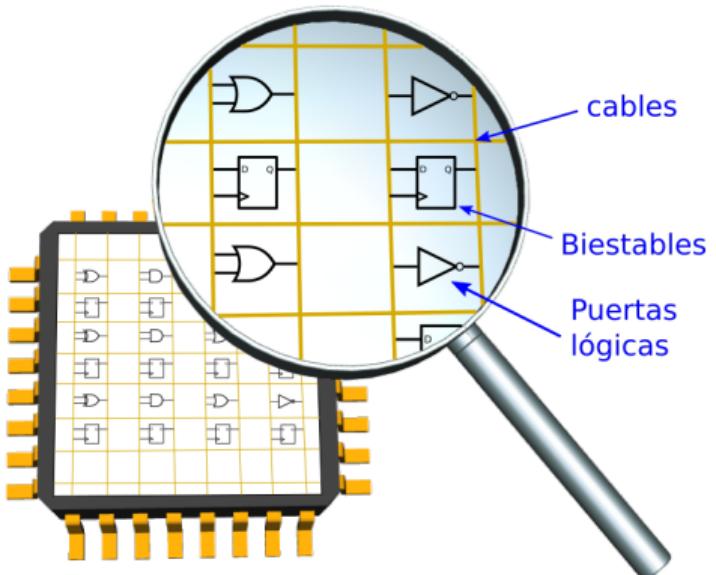
CIRCAE

ELEMSYS - Grupo de innovación



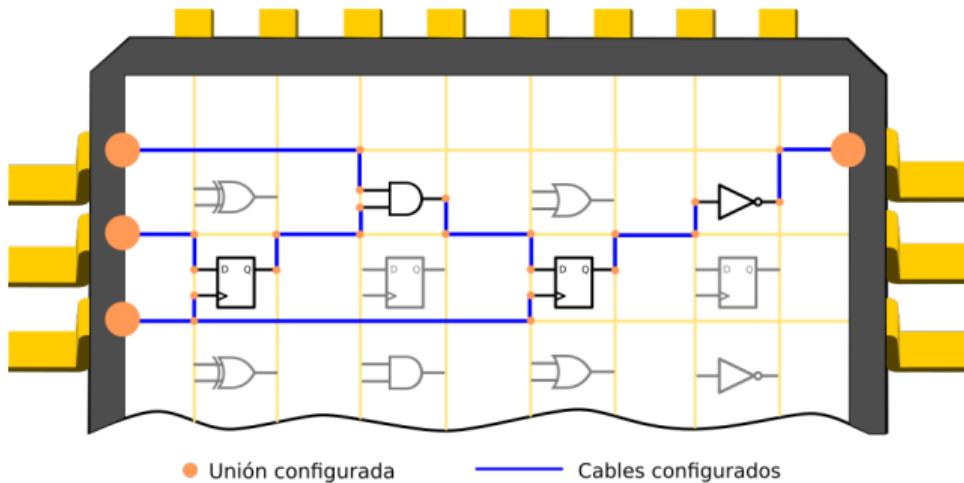
¿Qué es una FPGA?

- ▶ **FPGA: Field Programmable Gate Array** o “Arreglo de compuertas programables en campo”.
- ▶ Es un chip reconfigurable que permite implementar circuitos digitales personalizados.



Configuración de una FPGA

- ▶ Defines el comportamiento final del chip.
- ▶ Prototipar diseños para su posterior fabricación en un chip (ASIC).

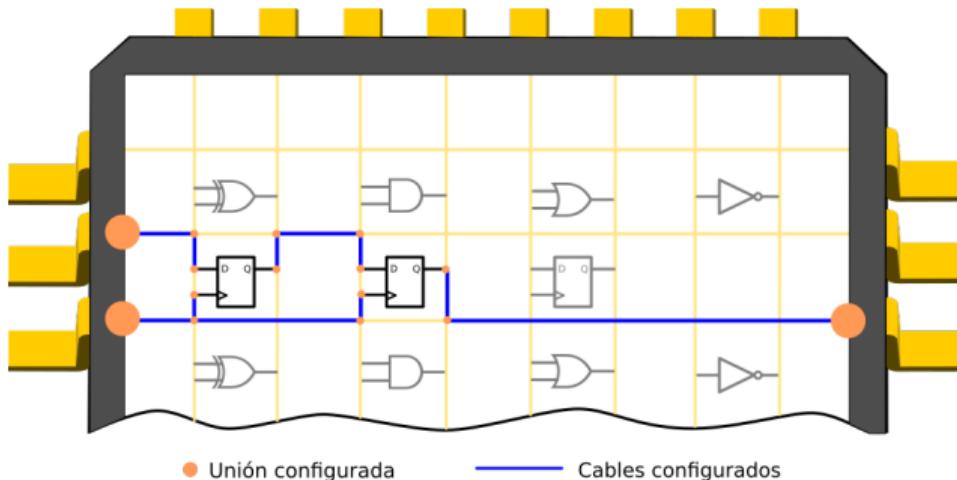


El hardware se diseña a medida

Otra configuración de la FPGA

- ▶ Internamente el chip se reconfigura para realizar otra tarea especificada por el usuario.
- ▶ En la imagen de abajo se muestra un divisor de frecuencia que genera a su salida:

$$f_{out} = \frac{f_{in}}{2(\# \text{ de FFs})} = \frac{f_{in}}{2^2}$$



Lenguajes de Descripción de Hardware (HDL)

Las FPGAs solamente entienden a los **lenguajes de descripción de hardware (HDL)** para implementar físicamente el diseño descrito.

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity compuerta_and is
5     port (
6         A : in std_logic;
7         B : in std_logic;
8         Salida : out std_logic
9     );
10 end entity;
11
12 architecture funcion of and_gate is
13 begin
14     Salida <= A and B;
15 end Behavioral;
```

VHDL

```
1 module compuerta_and(
2     input wire A,
3     input wire B,
4     output wire Salida
5 );
6
7 assign Salida = A & B;
8
9 endmodule
```

Verilog/System Verilog

Compuerta AND en VHDL

COMPUERTA AND		
A	B	Salida
0	0	0
0	1	0
1	0	0
1	1	1



```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity compuerta_and is
5     port (
6         A : in std_logic;
7         B : in std_logic;
8         Salida : out std_logic
9     );
10 end entity;
11
12 architecture funcion of and_gate is
13 begin
14     Salida <= A and B;
15 end Behavioral;
```

VHDL – Lenguaje muy usado en entornos académicos y aplicaciones críticas como aeroespacial, defensa, biomética y sistemas embebidos (estándar en la industria europea, académica y educativa).

Compuerta AND en Verilog

COMPUERTA AND		
A	B	Salida
0	0	0
0	1	0
1	0	0
1	1	1



```
1 module compuerta_and(  
2     input wire A,  
3     input wire B,  
4     output wire Salida  
5 );  
6  
7     assign Salida = A & B;  
8  
9 endmodule
```

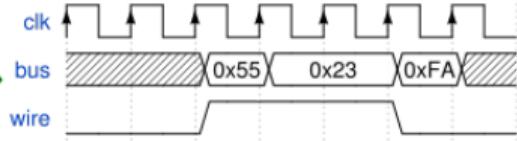
Verilog – Más común en la industria del diseño de chips y circuitos integrados, especialmente en Estados Unidos (**sector comercial y privado**).

Flujo de diseño

Fichero HDL

```
module simplez #(  
    parameter BAUD = 'B115200,  
    parameter WAIT_DELAY = 'T_200ms,  
    parameter ROMFILE = "prog.list",  
    parameter DEBUG_LEDS = 0  
  
);  
  
    input wire clk,  
    input wire rstn_ini,  
    output wire [3:0] leds,  
    output wire stop,  
    output wire tx,  
    input wire rx  
);  
  
    reg [DW-1: 0] alu_out;  
    reg flag_z;  
  
    always @(*) begin  
        if (alu_op2)  
            alu_out = alu_in;  
        else if (alu_clr)  
            alu_out = 0;  
        else if (alu_dec)  
            alu_out = reg_a + alu_in;  
        else if (alu_hex)  
            alu_out = reg_a - alu_in;  
    end  
endmodule
```

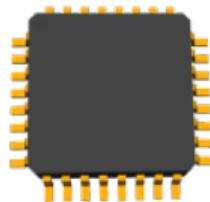
Simulación



Bitstream (FPGA)

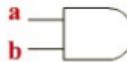
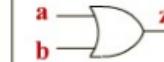
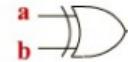
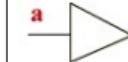
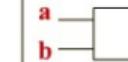
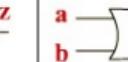
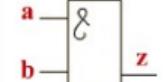
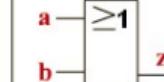
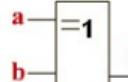
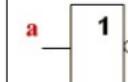
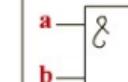
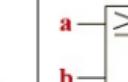
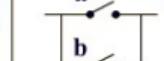
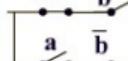
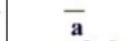
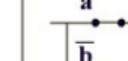
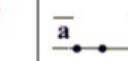
... 0110111100101 ...

Fabricación chips



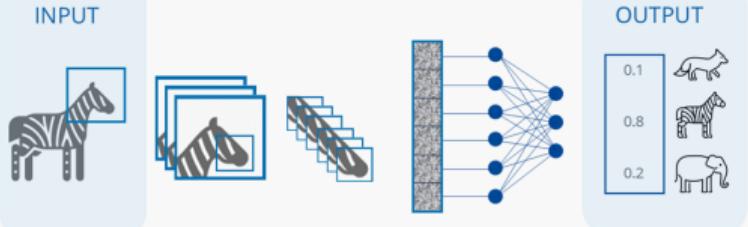
¿Puedo hacer cosas más complejas?

FUNCIONES LÓGICAS BÁSICAS

NOMBRE	AND - Y	OR - O	XOR O-exclusiva	NOT Inversor	NAND	NOR																																																																																	
SÍMBOLO																																																																																							
SÍMBOLO																																																																																							
TABLA DE VERDAD	<table border="1"> <tr><td>a</td><td>b</td><td>z</td></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 border="1"> <tr><td>a</td><td>b</td><td>z</td></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 border="1"> <tr><td>a</td><td>b</td><td>z</td></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 border="1"> <tr><td>a</td><td>z</td></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 border="1"> <tr><td>a</td><td>b</td><td>z</td></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 border="1"> <tr><td>a</td><td>b</td><td>z</td></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	$z = a \cdot b$	$z = a + b$	$z = \bar{a} \cdot b + a \cdot \bar{b}$	$z = \bar{a}$	$z = \overline{a \cdot b}$	$z = \overline{a + b}$																																																																																	

Red Neuronal Convolucional

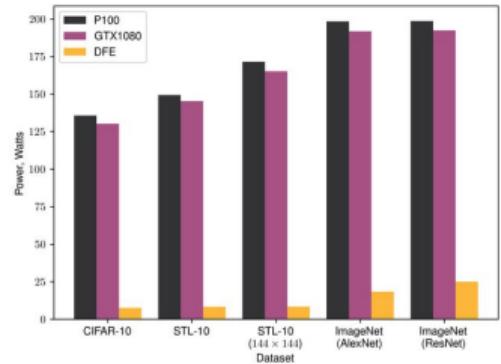
CONVOLUTIONAL NEURAL NETWORK (CNN)



```
1 library ieee;
2 use ieee.std_logic'1164.all;
3
4 entity red_neuronal is
5     port (
6         Capa_entrada : in std_logic;
7         Capa_oculta1 : in std_logic;
8             ...
9         Capa_oculta_n : in std_logic;
10        Capa_Salida : out std_logic
11    );
12 end entity;
13
14 architecture rtl of red_neuronal is
15 begin
16     (aquí tu propio diseño)
17 end Behavioral;
```

CPU/GPU vs FPGA

	CPU/GPU	FPGA
Reconfigurable	NO/SI	SI
Paralelismo real	NO/SI	SI
Control por software	SI/SI	SI
Consumo energético	Alto/Alto	Variable
Uso típico	PCs/IA	S. Embebidos

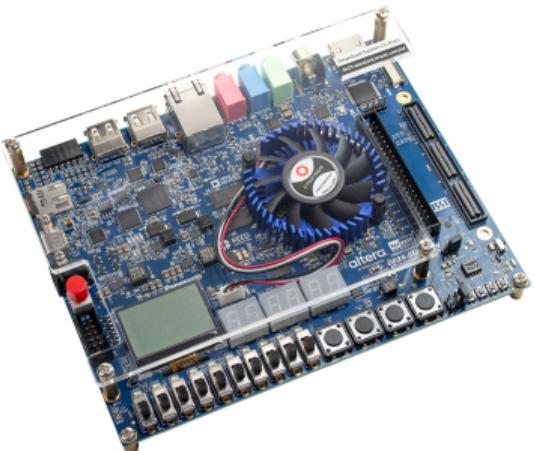
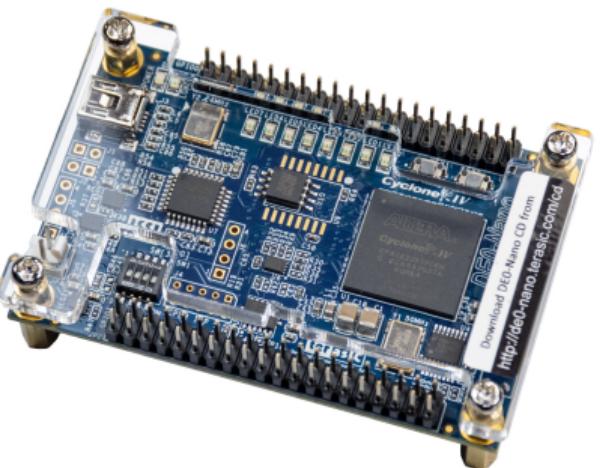


Fuente: FPGA-accelerated anisotropic diffusion filter based on SW/HW-codesign for medical images.

FPGAs actuales

► Placas comunes:

- ▶ Terasic DE10-Lite
- ▶ Digilent Basys 3



Entornos de desarrollo

► Software de desarrollo:

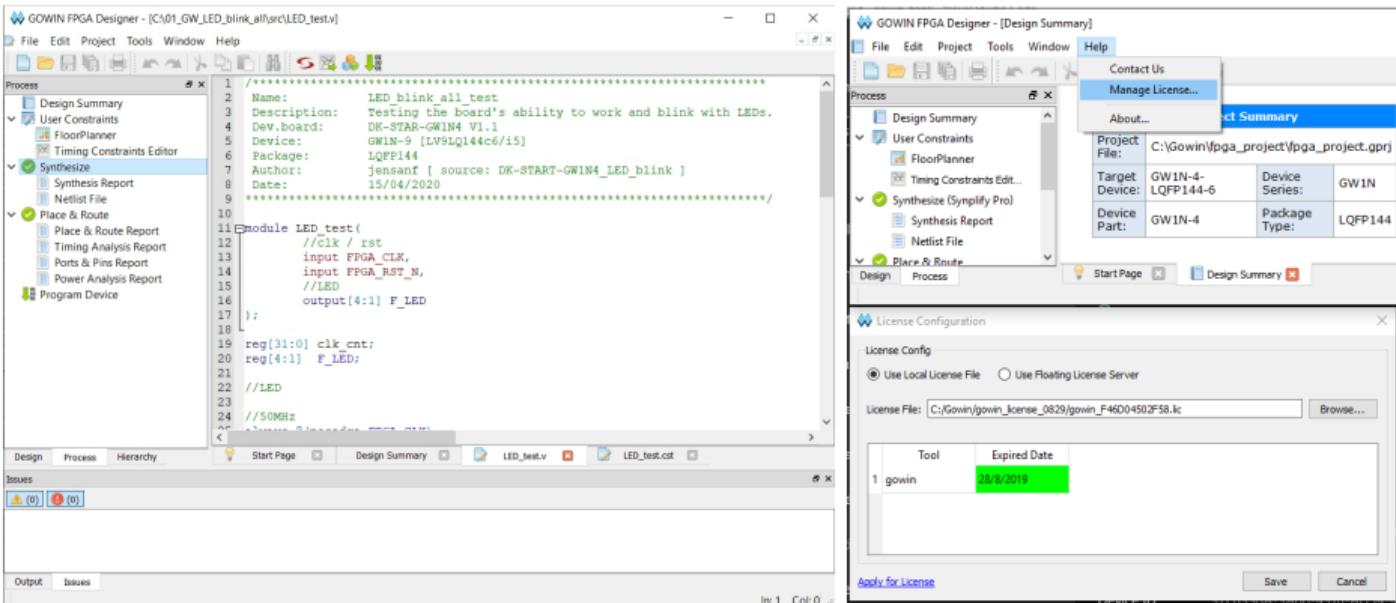
- Quartus, Vivado: muy potentes pero pesados.
- Requieren buen hardware y curva de aprendizaje alta.



Entorno de desarrollo Gowin IDE

Gowin IDE: entorno ligero y rápido

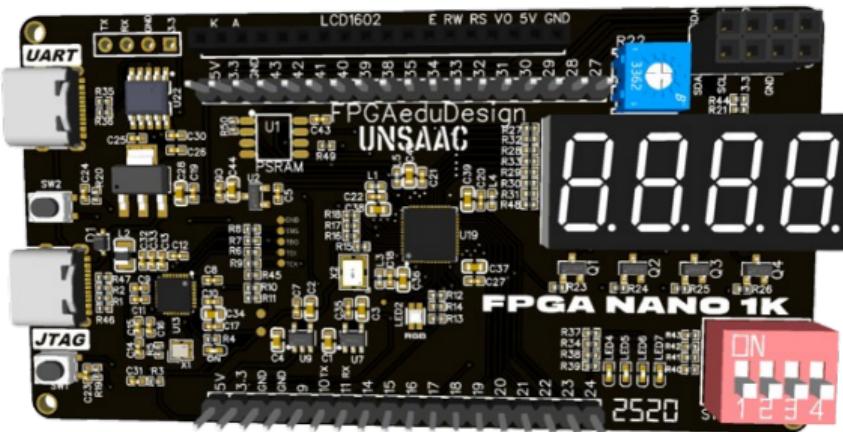
- ▶ Instalador < 1 GB
- ▶ Flujo de trabajo simplificado
- ▶ Ideal para iniciar en el mundo del hardware



Proyecto: FPGA NANO 1K

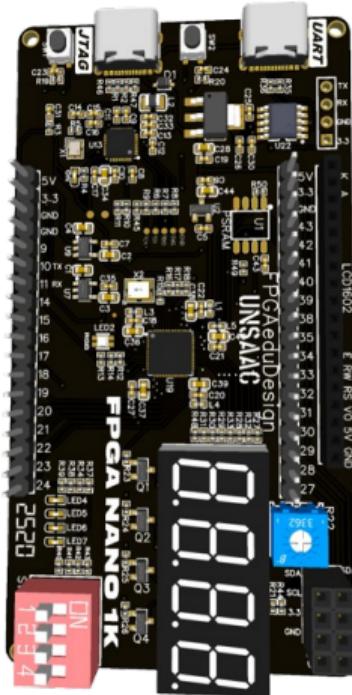
Un paso hacia la solución

- ▶ **Made in Perú/ Made in Cusco:** Diseñada y desarrollada por nuestro equipo como una herramienta educativa y experimental.
- ▶ Sistemas digitales básicos y avanzados.



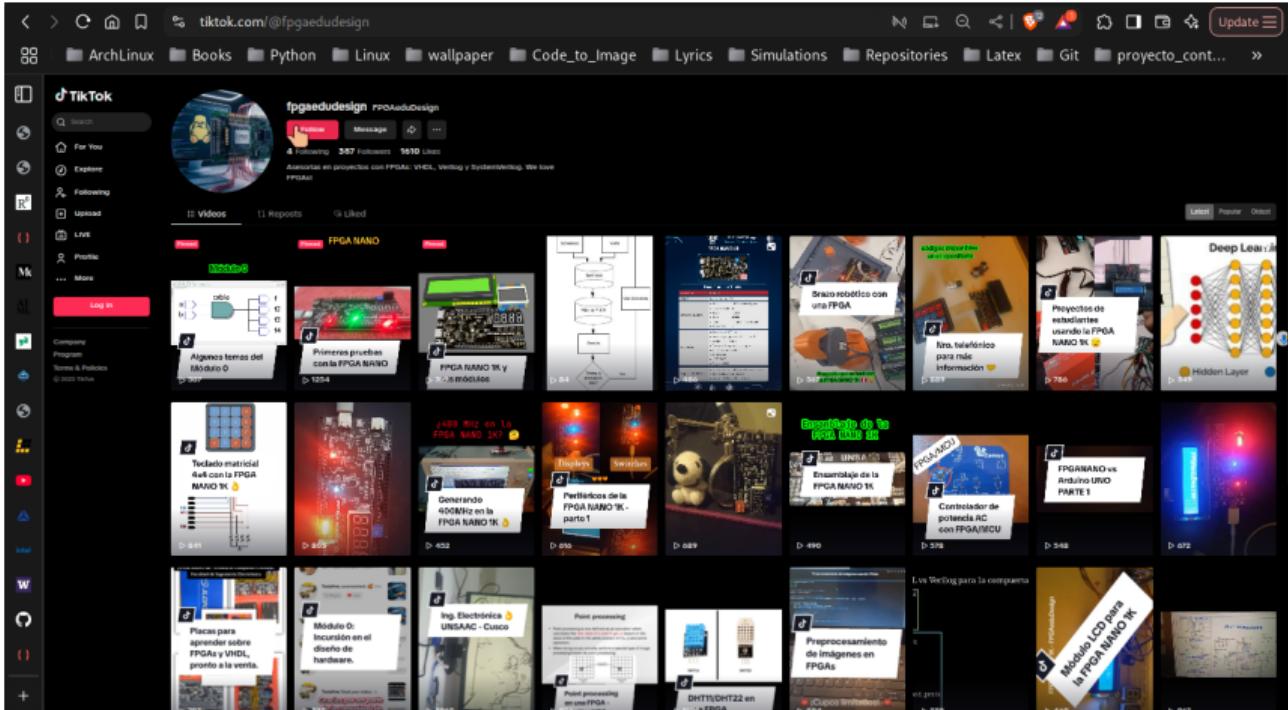
Arquitectura general de la FPGA NANO 1K

- ▶ FPGA Gowin/GW1NZ-LV1
- ▶ Entrada de reloj externo (27 MHz)
- ▶ 4 Switches y 7 LEDs
- ▶ 2 Botones de usuario
- ▶ Display de 8 segmentos (4 dígitos)
- ▶ Conector UART (TX/RX)
- ▶ 29 Pines GPIO para expansión
- ▶ Conector LCD 16x2



Placa de desarrollo FPGA NANO 1K.

Cuenta dedicada al uso de FPGAs



FPGAeduDesign

Se viene la FPGA NANO 9K

Posibles aplicaciones

- ▶ **Procesamiento de imágenes en tiempo real**
 - ▶ Adquisición de datos a frecuencias elevadas
 - ▶ Preprocesamiento: escala de grises, binarización, segmentación, mejorar calidad de imagen.
 - ▶ Operaciones matemáticas como convolución
 - ▶ Muchas otras cosas más...
- ▶ **Machine Learning en hardware**
 - ▶ Clasificación de colores para detección de plagas (roya en café).
 - ▶ Implementación de redes neuronales simples (inferencias en hardware).
- ▶ **Todo lo que puedas imaginar...pero**
- ▶ **Antes, requisito:** dominar todos los periféricos de la FPGA NANO 1K



“¿Te atreves a diseñar tu propio chip?”

Grupos Involucrados

Grupo de Investigación e innovación tecnológica: ELEMSYS

Centros de investigación: ELEMSYS, LIECAR, CIRCAE y BIOTRONIC

Redes Sociales

Tiktok: ELEMSYS | FPGAeduDesign

Facebook: ELEMSYS | FPGAeduDesign

Youtube: ELEMSYS | FPGAeduDesign