

FPGAs libres



Juan González Gómez (Obijuan)

<https://github.com/Obijuan>

Circuitos digitales

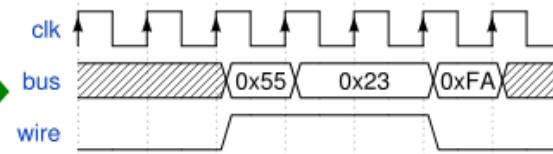
Fichero HDL

```
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;
    //-- Suma de operador 1 + operador 2
    else if (alu_add)
        alu_out = reg_a + alu_in;
    else if (alu_dec)
        alu_out = reg_a - alu_in;
end

```

Simulación

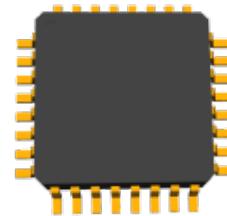


Bitstream (FPGA)

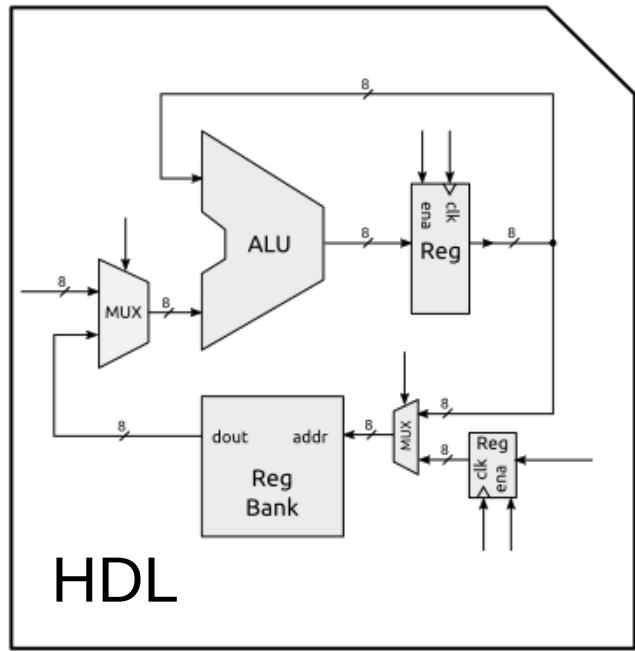
... 0110111100101 ...



Fabricación chips



El hardware es software



Bitstream

... 0110111100101 ...

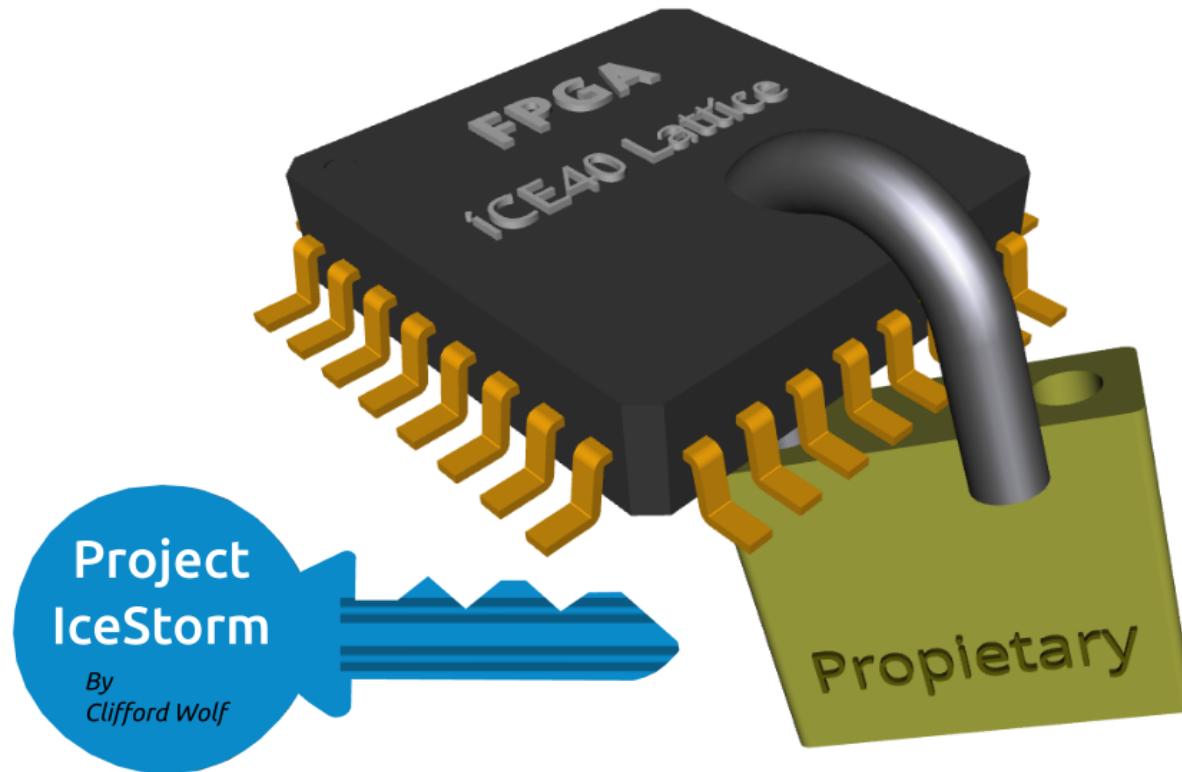
- El hardware libre es igual al software libre
- Muy fácil de compartir
- Telecopias del hardware
- Desarrollo de hardware en comunidad

FPGAs: Sólo personal autorizado



- Sólo el fabricante conoce los detalles internos
- Sólo se puede usar lo que el fabricante haya previsto
- Atados de por vida al fabricante

FPGAs libres: El renacimiento



- Proyecto Icestorm (Mayo, 2015)
- La primera *toolchain* que permiten pasar de Verilog al bitstream usando sólo Herramientas libres

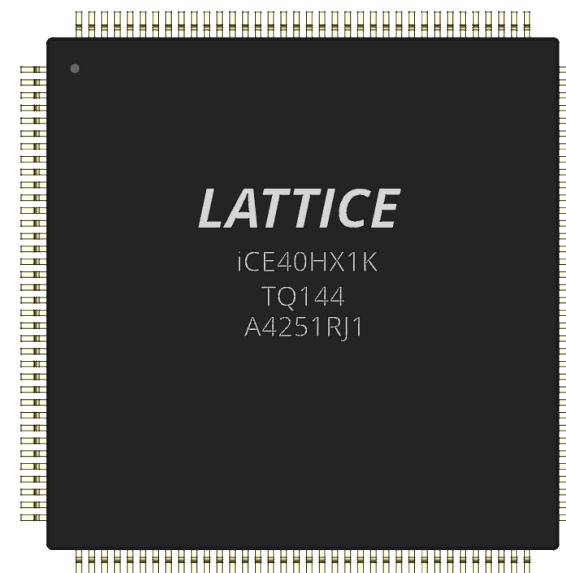
FPGAs libres

- Definición:

Denominamos **FPGAs libres** a aquellas FPGAs que disponen de una **toolchain totalmente libre**

- **FPGAs libres actualmente:**

- Familia **Lattice iCE40**
- Sólo Lenguaje Verilog



<http://www.latticesemi.com/Products/FPGAandCPLD/iCE40.aspx>

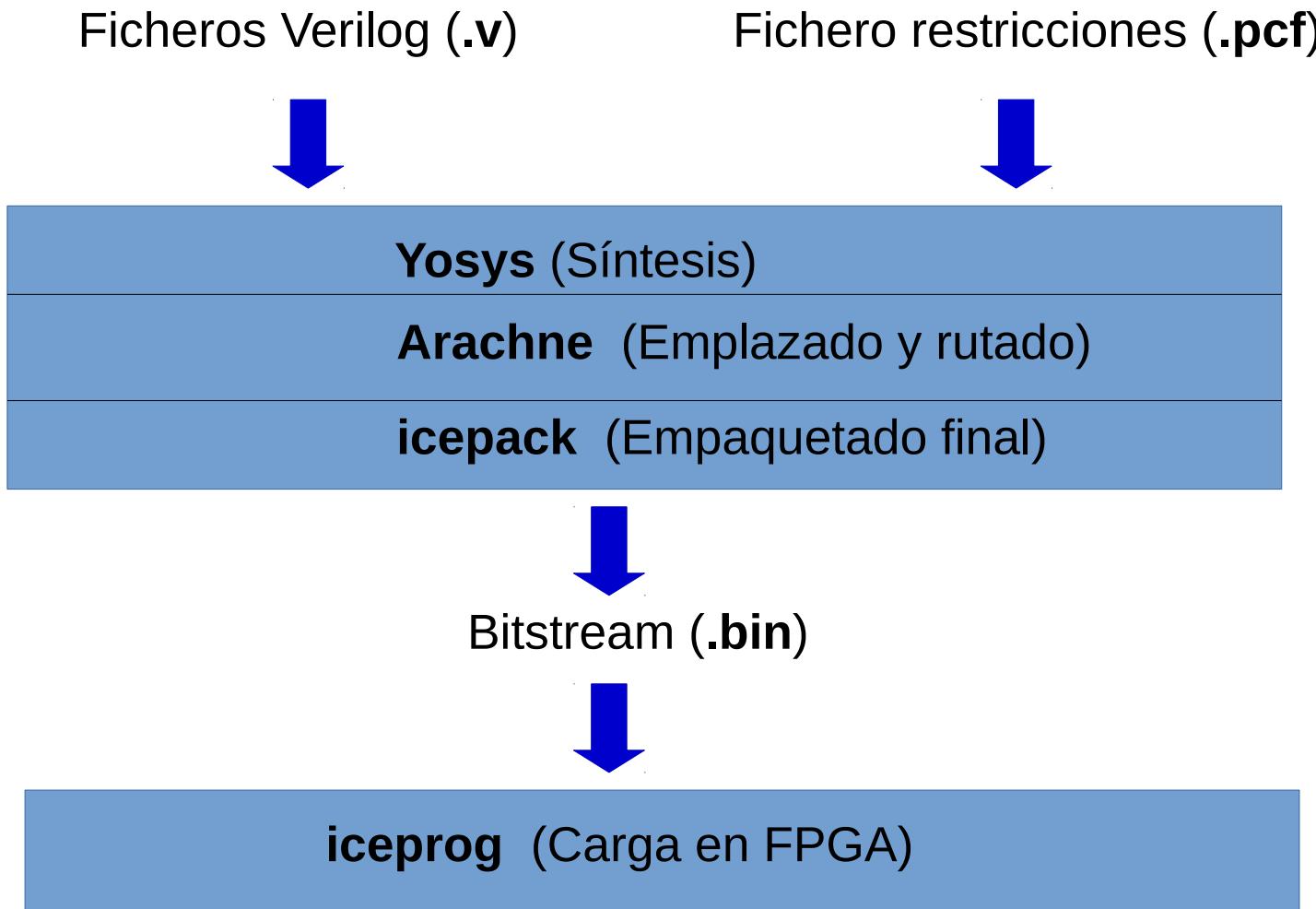
Proyecto Icestorm

- Herramientas programadas en C/C++
- Bajar del repo y compilar
- Línea de comandos
- Se usan típicamente junto con make
- (bajo nivel)

<http://www.clifford.at/icestorm/>

<https://github.com/cliffordwolf/icestorm>

Proyecto Icestorm: Herramientas



Proyecto Icestorm: Ejemplo

- **Ficheros:** *cont8.v icezum.pcf*
- **Síntesis:**

Demo

```
$ yosys -p "synth_ice40 -blif cont8.blif" cont8.v
```

- **Emplazado y rutado:**

```
$ arachne-pnr -d 1k -p icezum.pcf cont8.blif -o cont8.asc
```

- **Empaquetado:**

```
$ icepack cont8.asc cont8.bin
```

- **Carga en FPGA:**

```
$ iceprog cont8.bin
```



APIO

- Autor: **Jesús Arroyo**
- Multiplataforma (Linux, Mac, Windows, Raspberry)
- Línea de comandos
- Programado en python
- Multiplaca: icestick, icezum, icoboard, go-board

Comandos



APIO

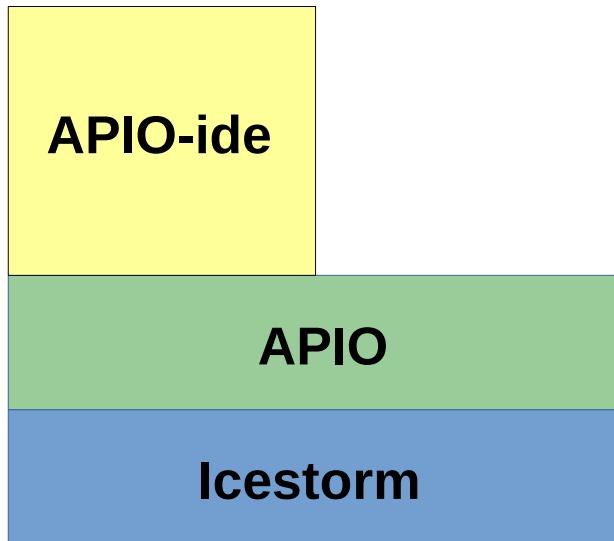
Icestorm

<https://github.com/FPGAwars/apio>

Demo



APIO-ide



- Plug-in para Atom
- No línea de comandos
- Llama a apio
- Aplicable a otros IDEs/editores
- Descripción en Verilog

<https://github.com/FPGAwars/apio-ide>

APIO ide

The screenshot shows the APIO ide interface. On the left is a file tree for a project named "contador-8-bits-apio". The "main.v" file is selected and open in the main editor area. The code is a Verilog module named "top" that defines a 8-bit counter with a prescaler of 20. The code includes comments explaining the purpose of each part: the prescaler bits, the counter register, and the assignment of the most significant 8 bits of the counter to the LED outputs.

```
//--- Contador de 8 bits con prescaler
/// Parametro N: Numero de bits para usar en el prescaler
/// Board: icezum
`default_nettype none

module top(input wire CLK,
            output wire LED0,
            output wire LED1,
            output wire LED2,
            output wire LED3,
            output wire LED4,
            output wire LED5,
            output wire LED6,
            output wire LED7);

    parameter N = 20;

    reg [N-1 + 8:0] counter = 0;

    always @ (posedge CLK) begin
        counter <= counter + 1;
    end

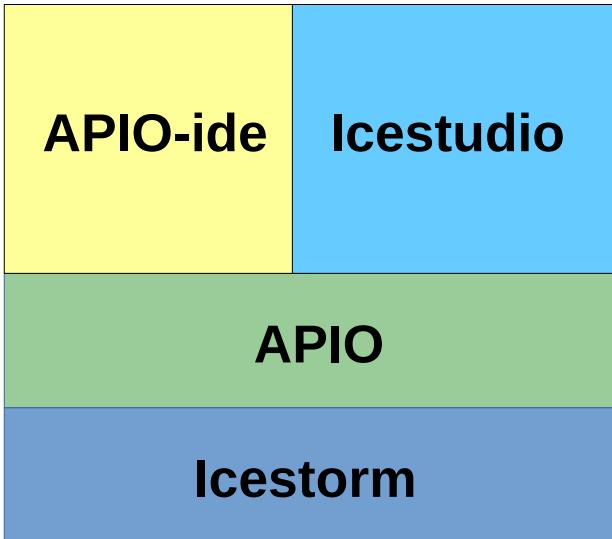
    //-- Sacar los 8 bits mas significativos como salida del contador por los leds
    assign {LED7, LED6, LED5, LED4, LED3, LED2, LED1, LED0} = counter[7+N:N];
endmodule
```

At the bottom of the interface, there are tabs for "File 0", "Project 0", "No Issues", "main.v", "1:1", "LF", "UTF-8", "Verilog", "master", "+10,-10", and a settings gear icon.

Demo



icestudio



- Autor: **Jesús Arroyo**
- Electrónica digital para todos
- Sin conocimientos de verilog
- Herramienta visual
- Traduce a verilog

<https://github.com/FPGAwars/icestudio>

Icestudio: Demo

3_switch_and_gate - Icestudio

File ▾ Edit ▾ View ▾ Boards ▾ Tools ▾ Help ▾ Basic ▾ Bit ▾ Config ▾ Logic ▾

```
graph LR; button1[button1 SW1 x] --> inv1[Inverter]; button2[button2 SW2 x] --> inv2[Inverter]; inv1 --> and[AND]; inv2 --> and; and --> led[led LED7 x]
```

Basic AND gate circuit
A 2-inputs AND logic gate is used to turn on the LED7 only when the 2 input buttons are pressed
This example shows the basic behaviour of the AND gate
EXERCISE: Upload this circuit into the FPGA board and play with it

Circuito básico con puerta AND
Una puerta lógica AND de 2 entradas se usa para encender un led solo cuando los dos pulsadores de entrada están apretados
Este ejemplo muestra el comportamiento básico de una puerta AND
EJERCICIO: Carga este circuito en la FPGA y juega con él

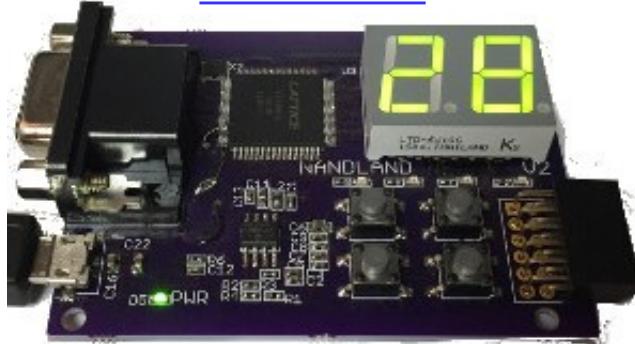
3_switch_and_gate

Tarjetas entrenadoras con FPGAs libres

[Icestick](#)



[Go-board](#)



[iCE40-HX8K Breakout Board](#)



[Icezum Alhambra](#)



- Conexión directa al PC (USB)
- Soportadas por Apio/Icestudio

Tarjetas entrenadoras con FPGAs libres

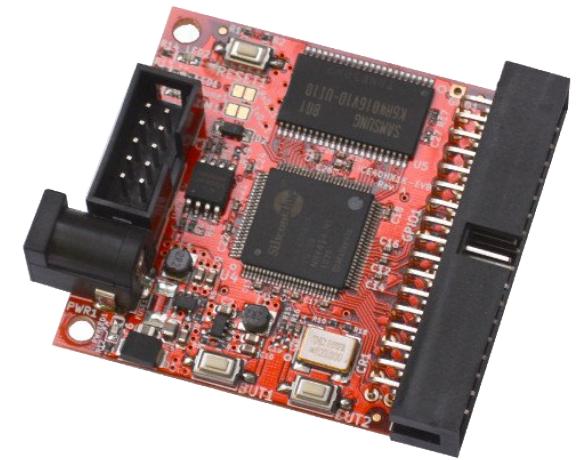
[icoboard](#)



[Mystorm](#)



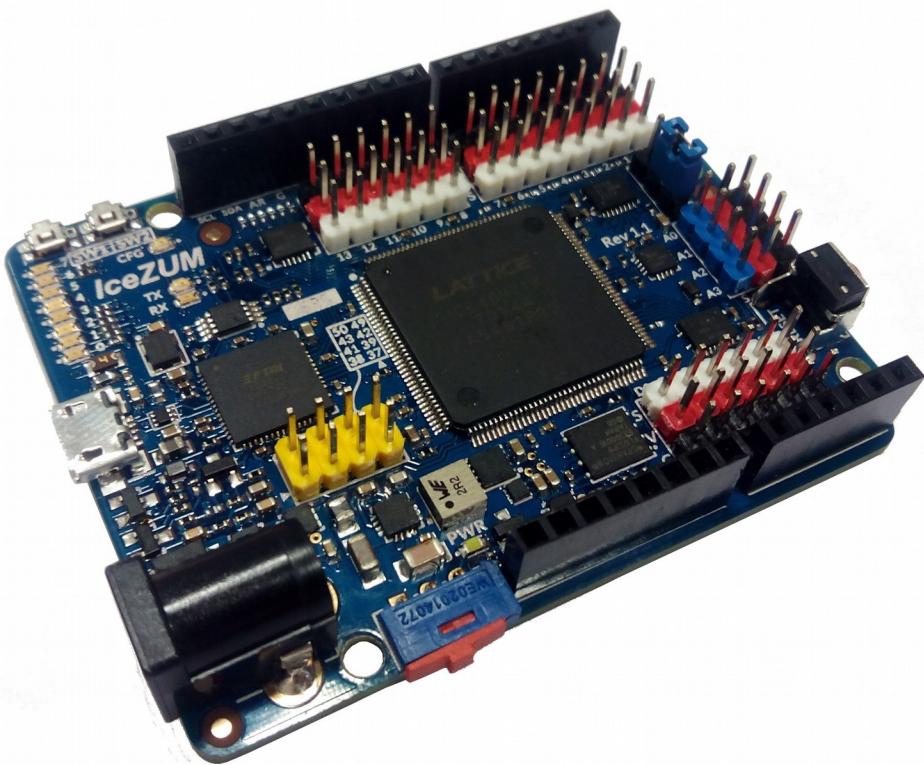
[iCE40HX1K-EVB](#)



- Conexión a Raspberry PI
- Soportada por Apio/Icestudio

NO Soportadas por Apio/Icestudio

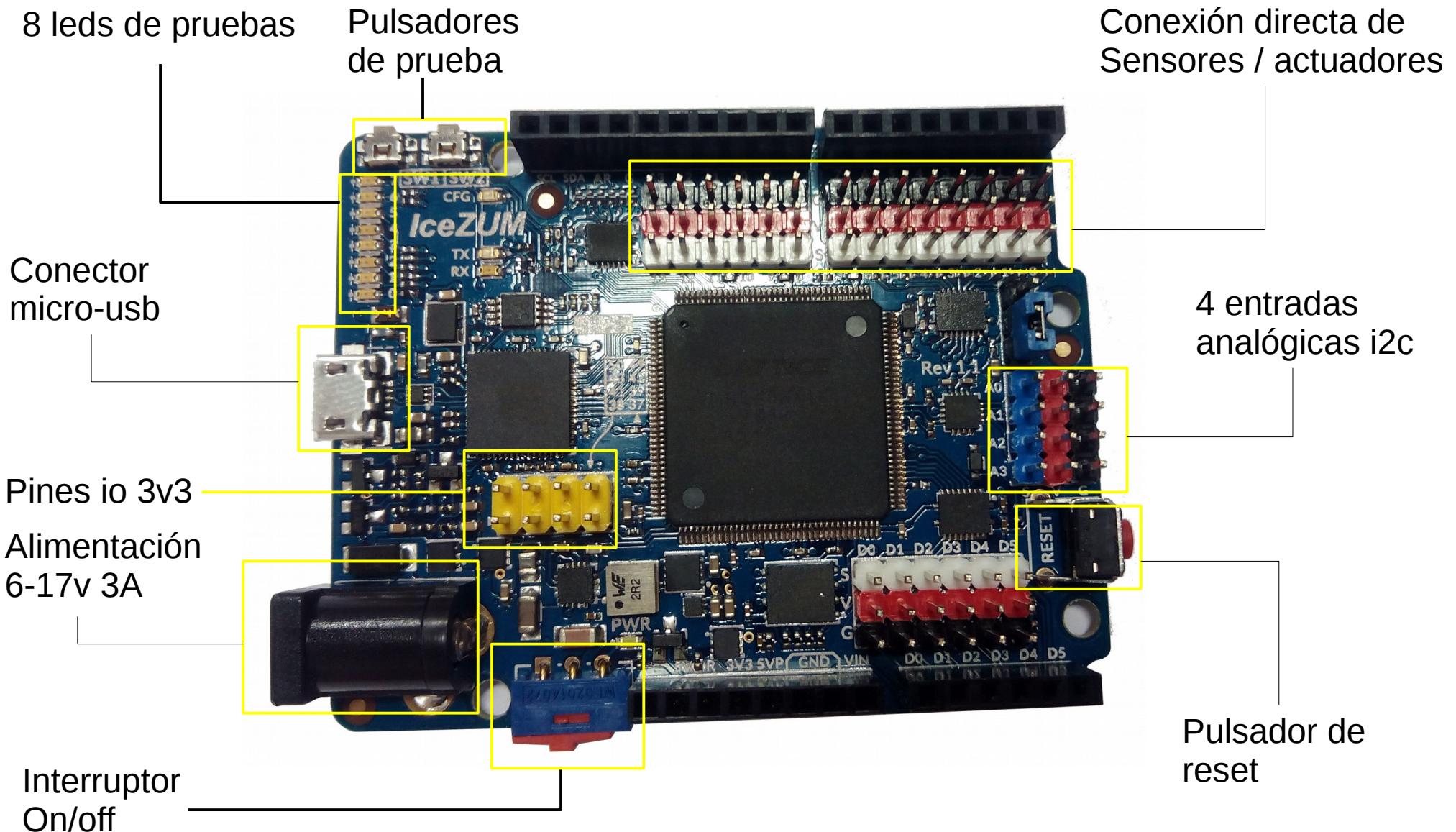
Icezum Alhambra v1.1



- Autor: **Eladio Delgado**
- Diseñada en Pinos del Valle (Granada)
- Arduino de las FPGAs
- Compatible Arduino
- Fácil conexión de circuitos externos/sensores/servos
- Reutilización de los shields de arduino
- 20 entradas/salidas de 5v
- 3A corriente de entrada
- Perfecta para hacer robots

<https://github.com/FPGAwars/icezum/wiki>

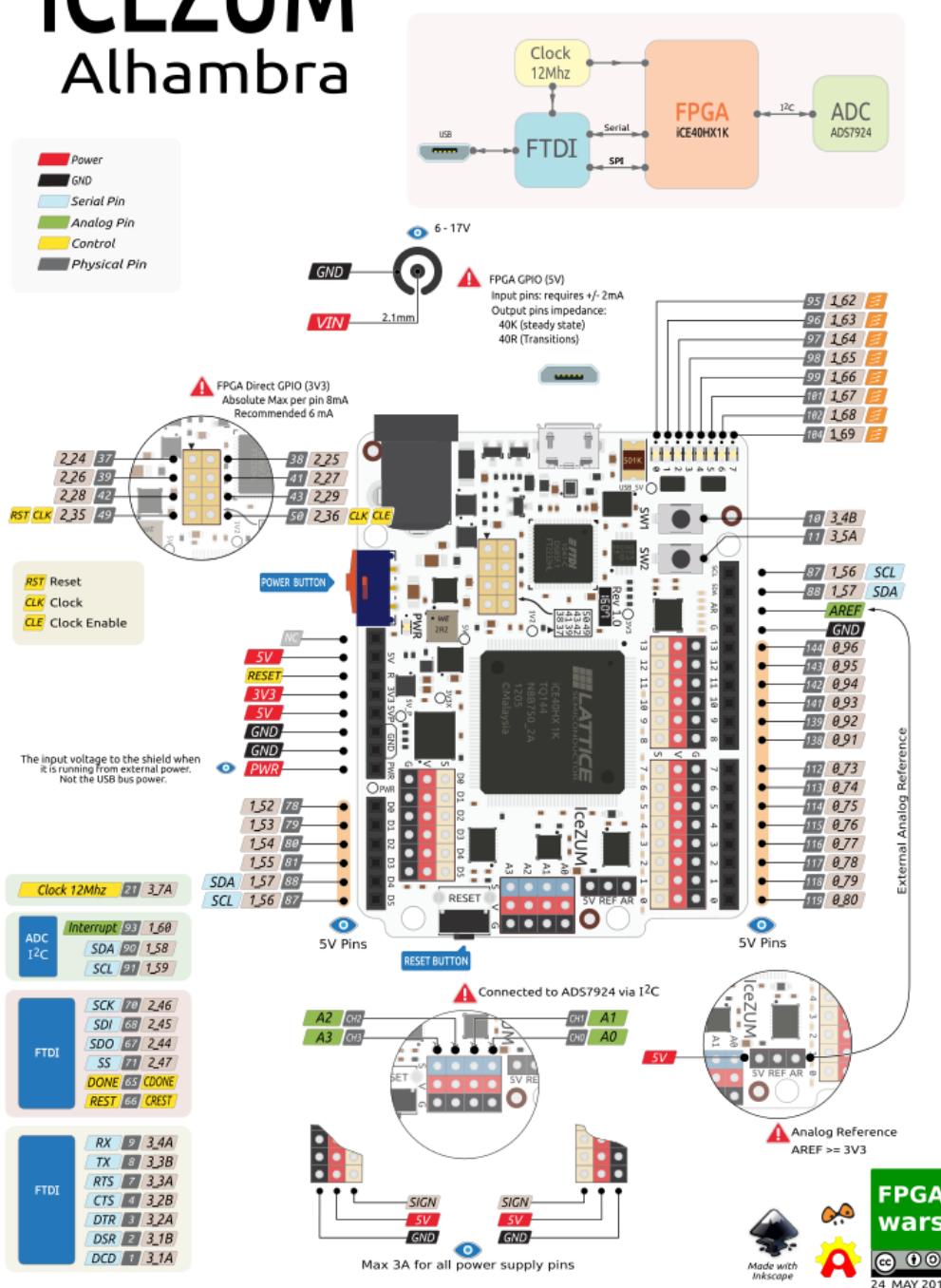
Icezum Alhambra v1.1



iCEZUM

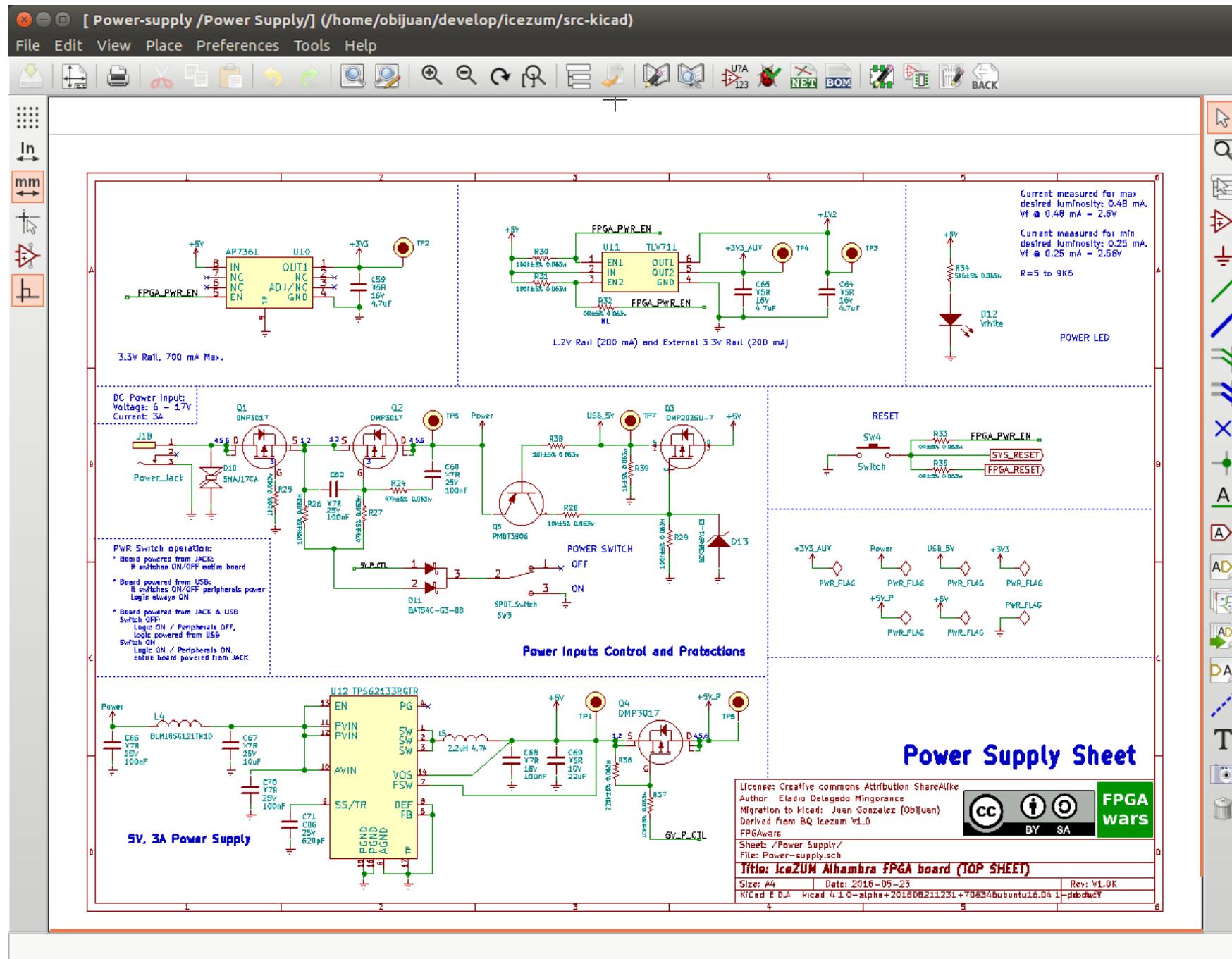
Alhambra

Power
 GND
 Serial Pin
 Analog Pin
 Control
 Physical Pin

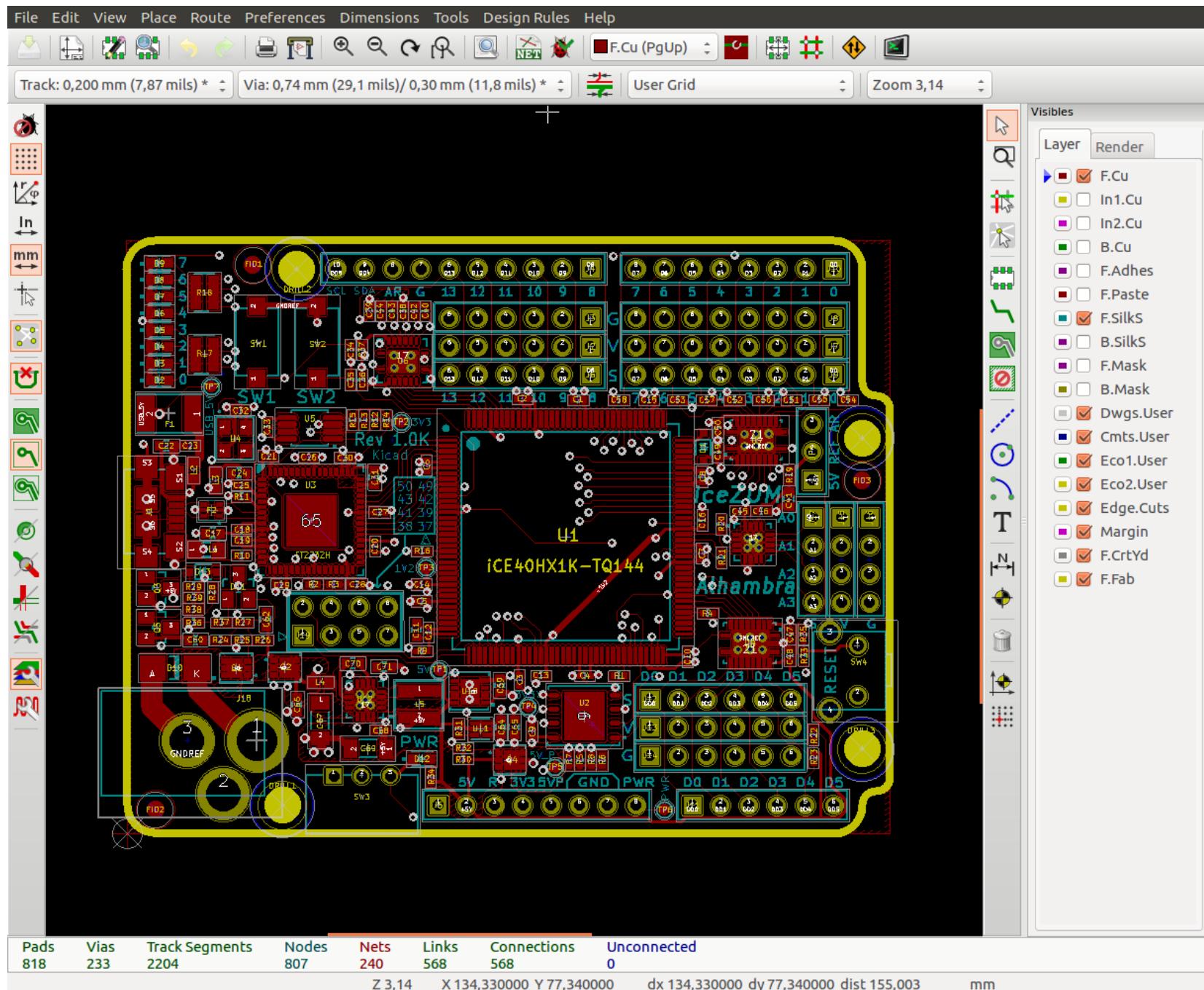


Pinout: by Alberto Piganti

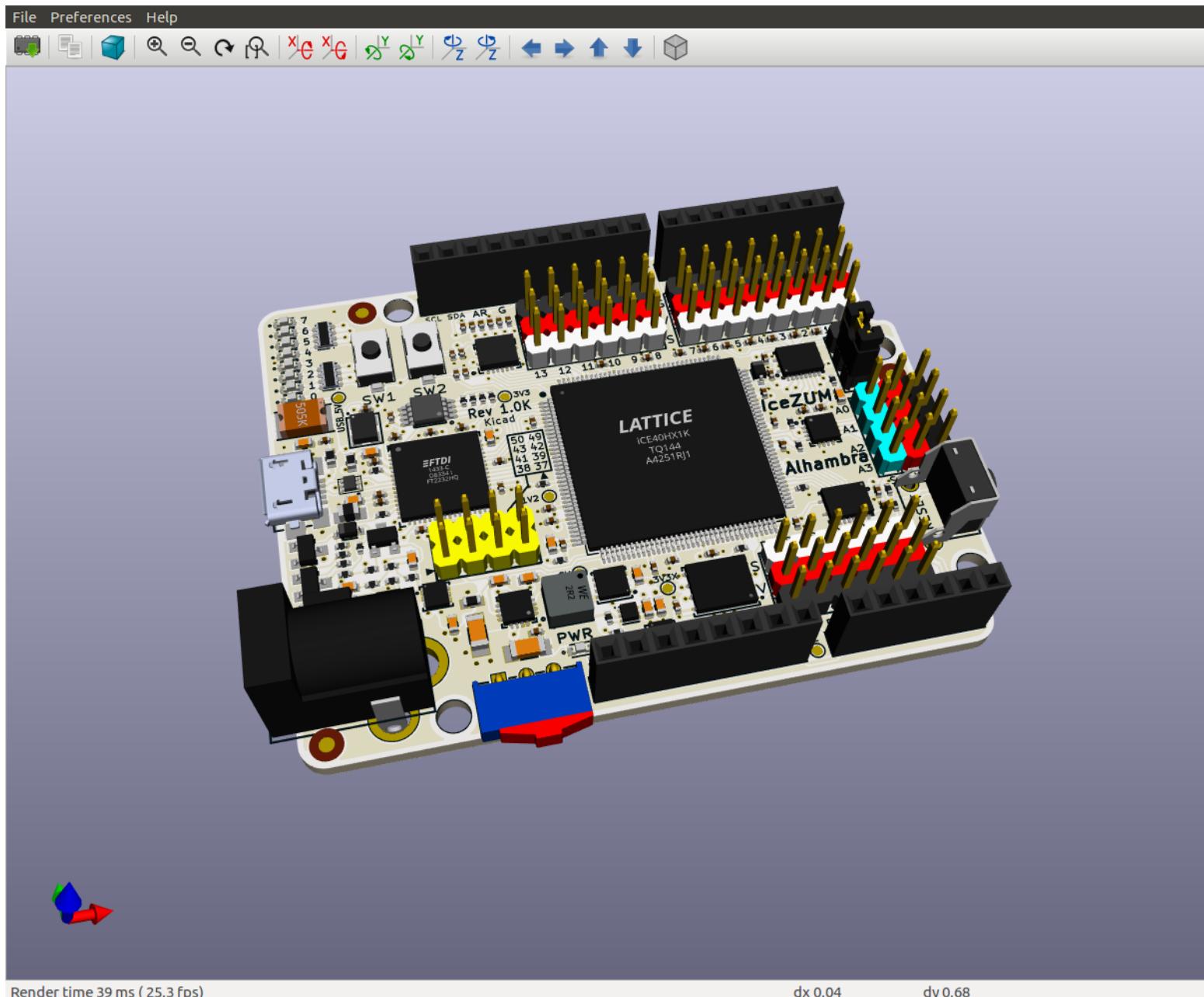
Icezum Alhambra v1.0K en Kicad



Icezum Alhambra v1.0K en Kicad



Icezum Alhambra v1.0K en Kicad



Demo

Comunidad FPGAwars



- Comunidad para **compartir conocimiento** relacionado con **FPGAs libres**
- Es el **clonewars** de las FPGAs, pero en modesto :-)
- Idioma: Castellano
- 150 miembros
- Cualquier pregunta / comentario / sugerencia → Correo a la lista :-)

<http://fpgawars.github.io/>

Me molan las FPGAs libres... ¿Por dónde empiezo?

Paso 1: Consigue una placa con FPGA libre

Icestick



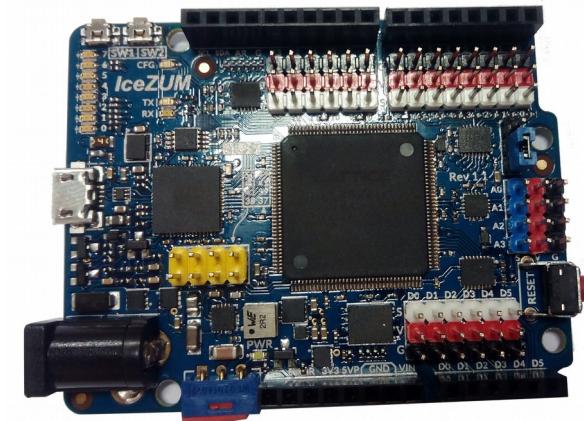
Go-board



iCE40-HX8K Breakout Board



Icezum Alhambra

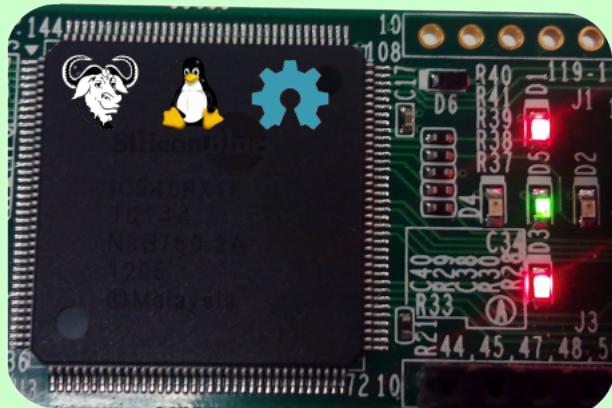


Me molan las FPGAs libres... ¿Por dónde empiezo?

Paso 2: Instálate Apio / Icestudio

Paso 3: Empieza a aprender verilog

[Tutorial: Diseño Digital para FPGAs, con herramientas libres](#)



Me molan las FPGAs libres... ¿Por dónde empiezo?

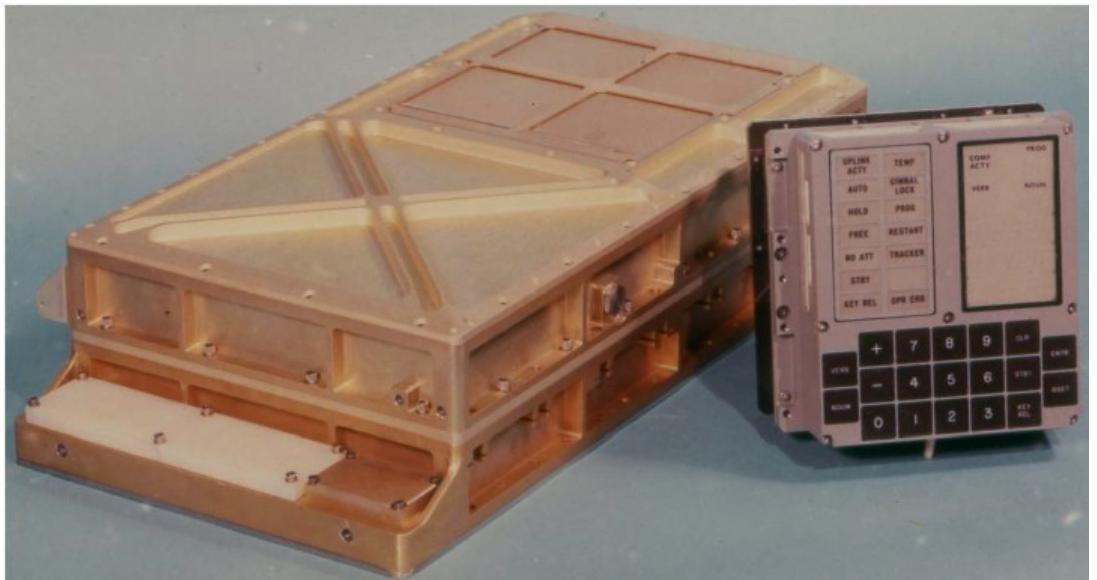
Paso 4: Apúntate al grupo de FPGAwars y haz preguntas

- Las FPGAs libres evolucionan muy rápido y en seguida la información se queda obsoleta
- En FPGAwars es donde se cuece lo último de lo último

Paso 5: Haz tus propios proyectos con FPGAs libres

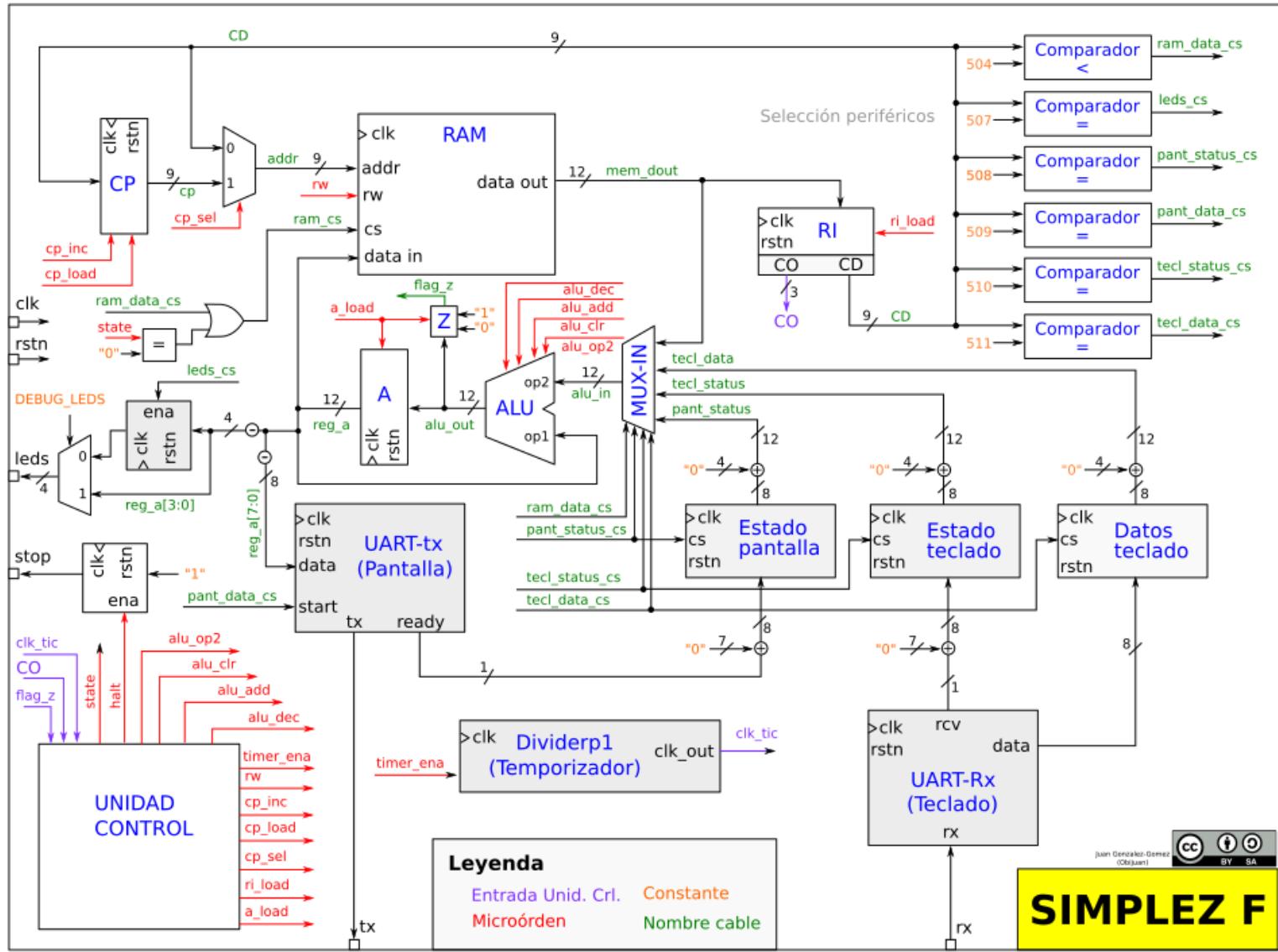
Paso 6: ¡Comparte tu proyecto con la comunidad! :-)

Proyecto ACC Apollo CPU Core



Implementación de la CPU del Apollo en FPGAs libres

<https://github.com/Obijuan/ACC/wiki>

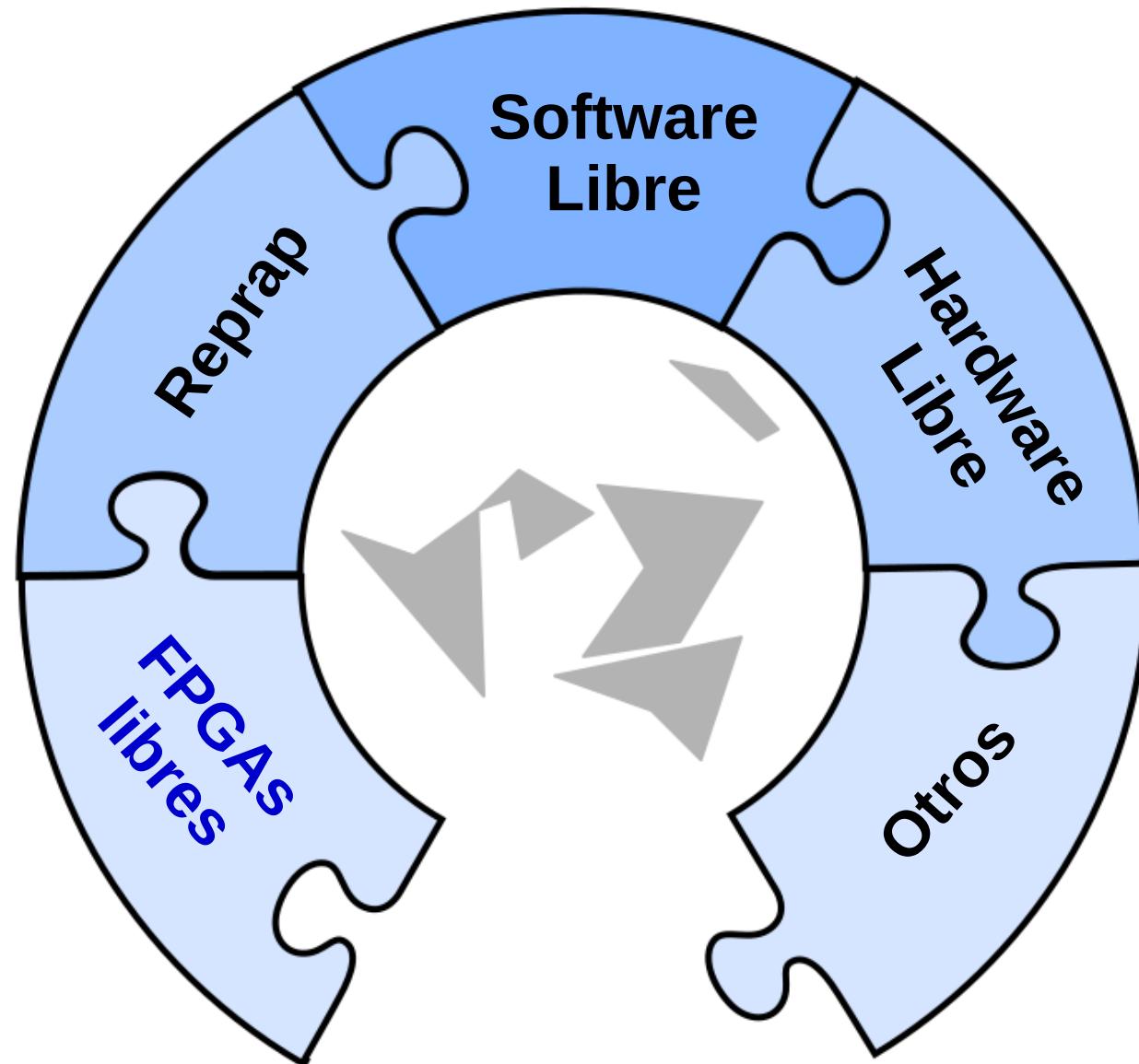


Procesador educativo Simplez para FPGAs libres
<https://github.com/Obijuan/simplez-fpga/wiki/Procesador-SIMPLEZ-F>

CC BY SA

SIMPLEZ F

FPGAs en Patrimonio Tecnológico de la humanidad



FPGAs libres



Juan González Gómez (Obijuan)

<https://github.com/Obijuan>