

# Herramientas Open-Source para desarrollo de FPGAs

GDG Gasteiz  
2022/05/20  
Iñigo Muguruza

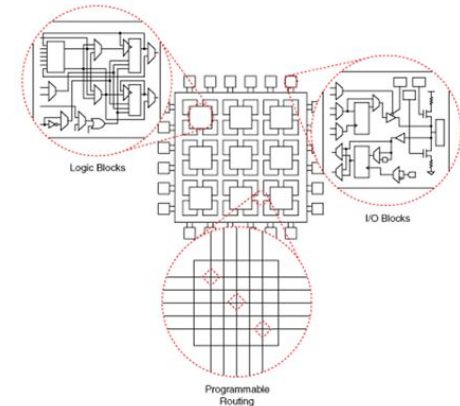
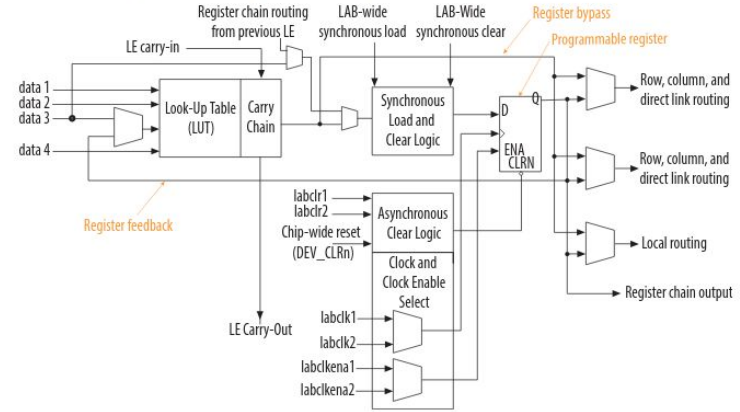
# Agenda

- Intro FPGAs - uso, que son, campos, workflow, herramientas
- F4PGA, el “gcc” de las FPGAs
- Ecosistema open-software/hardware para FPGAs
- Live Demo
- Extras - Links y recursos

# Intro - ¿Cómo son?

- Suele contener:
  - Interconexiones
  - Logic Elements o CLB
  - DPSs
  - Block RAM
  - PLLs
  - Transceivers PCIe, ....
  - Hard IPs especializados: por ejemplo ADCs
  - Pines de IO

Figure 3. Intel® Cyclone® 10 LP Device Family LEs



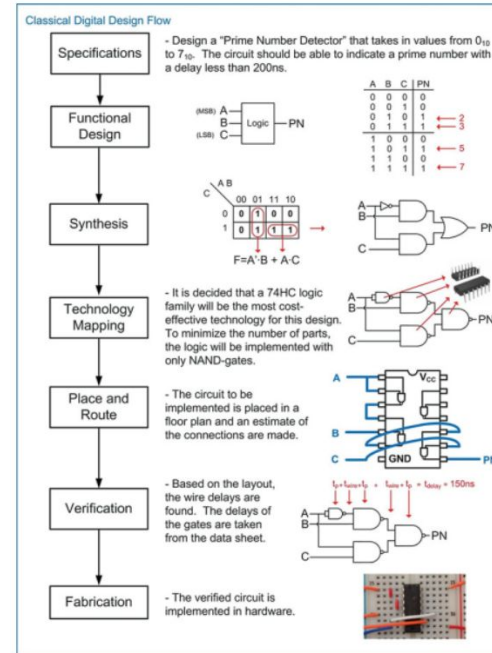
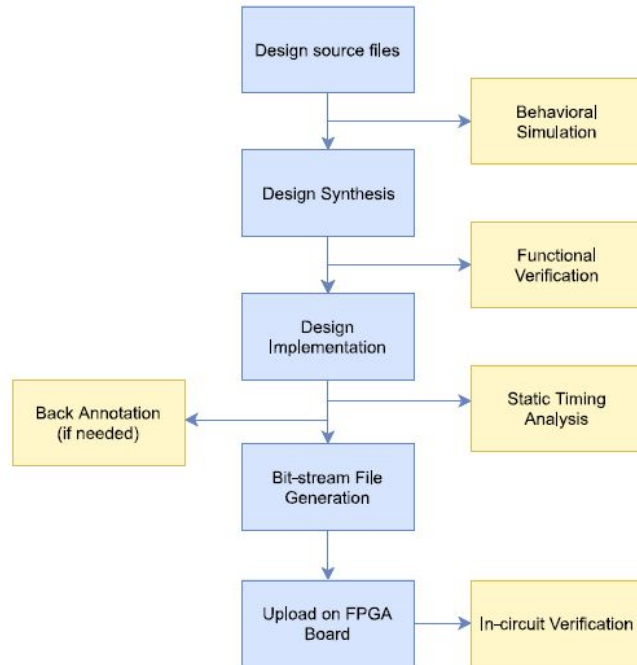
# Intro

- Las FPGAs están de moda de nuevo:
  - Mercados nuevos: AI, robotica, coches autonomos, investigación sobre nuevas arquitecturas de computación, emuladores de consolas implementadas en hw (doom), ...
- Pero siempre han estado ahí:
  - Mercados clasicos: electrónica de potencia, ferroviario, espacio y defensa, procesamiento de audio y video, criptografia, desarrollo de ASICs, data-centers o software-defined radio.
- La filosofía del mundo software ha entrado también en este campo
  - Desarrollos open-source (por ejemplo opencores) y herramientas open-source (iverilog, verilator, gtkwave, f4pga o cpu cores y RISC-V).

# Intro

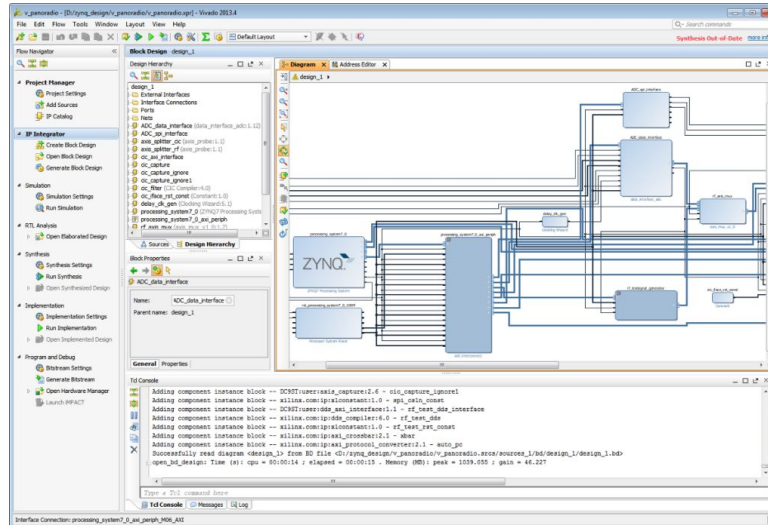
- Ventajas:
  - Verdadero paralelismo de computación
  - Ad-hoc a la aplicación
  - Computación/Watts vs CPUs y GPUs
  - Reconfigurables (son como un lienzo) parcial o total
  - “Real-Time”
- Desventajas
  - La principal, curva de aprendizaje => pensar en lógica digital
  - Típicamente, se programan en hdl (hardware description language), no es lo mismo que C/Python/C++, lógica digital.

# Intro - Workflow



# Intro - Herramientas

- Intel (Altera) y AMD/Xilinx los fabricantes más grandes de FPGAs
- Intel => Quartus Prime
- Xilinx => Vivado y Vitis
- Renesas ha anunciado FPGAs, parece que con herramientas open-source, [link](#).

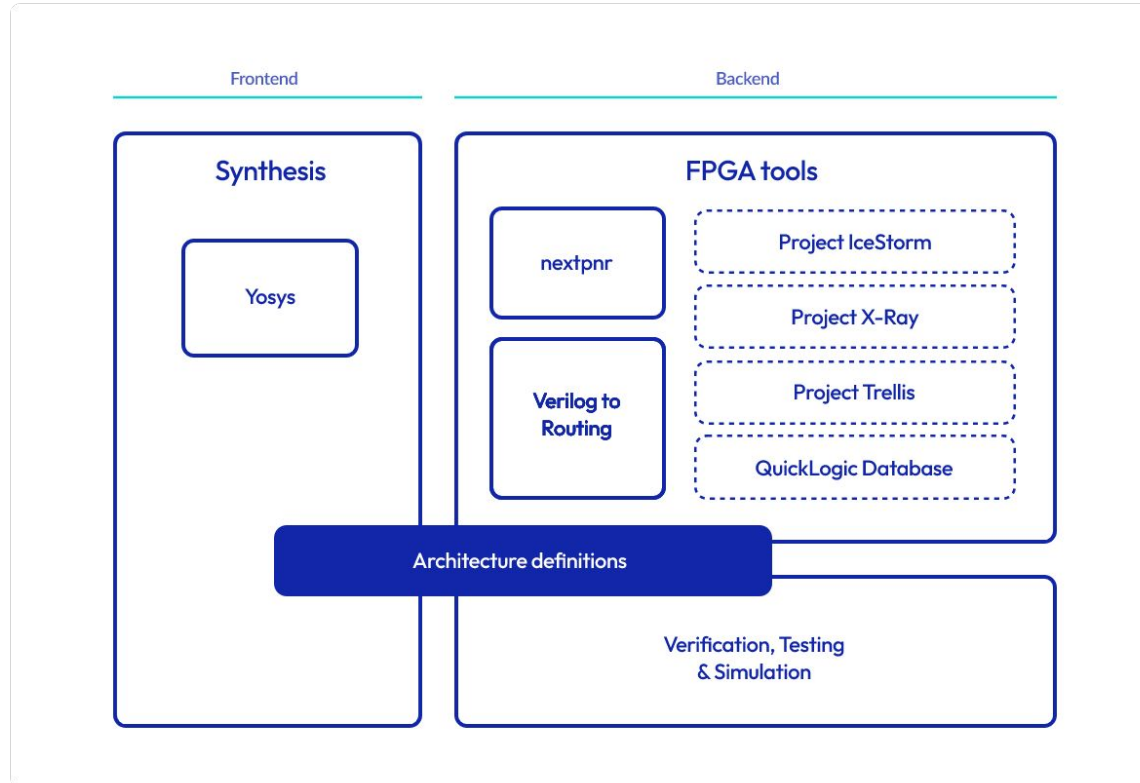


# F4PGA

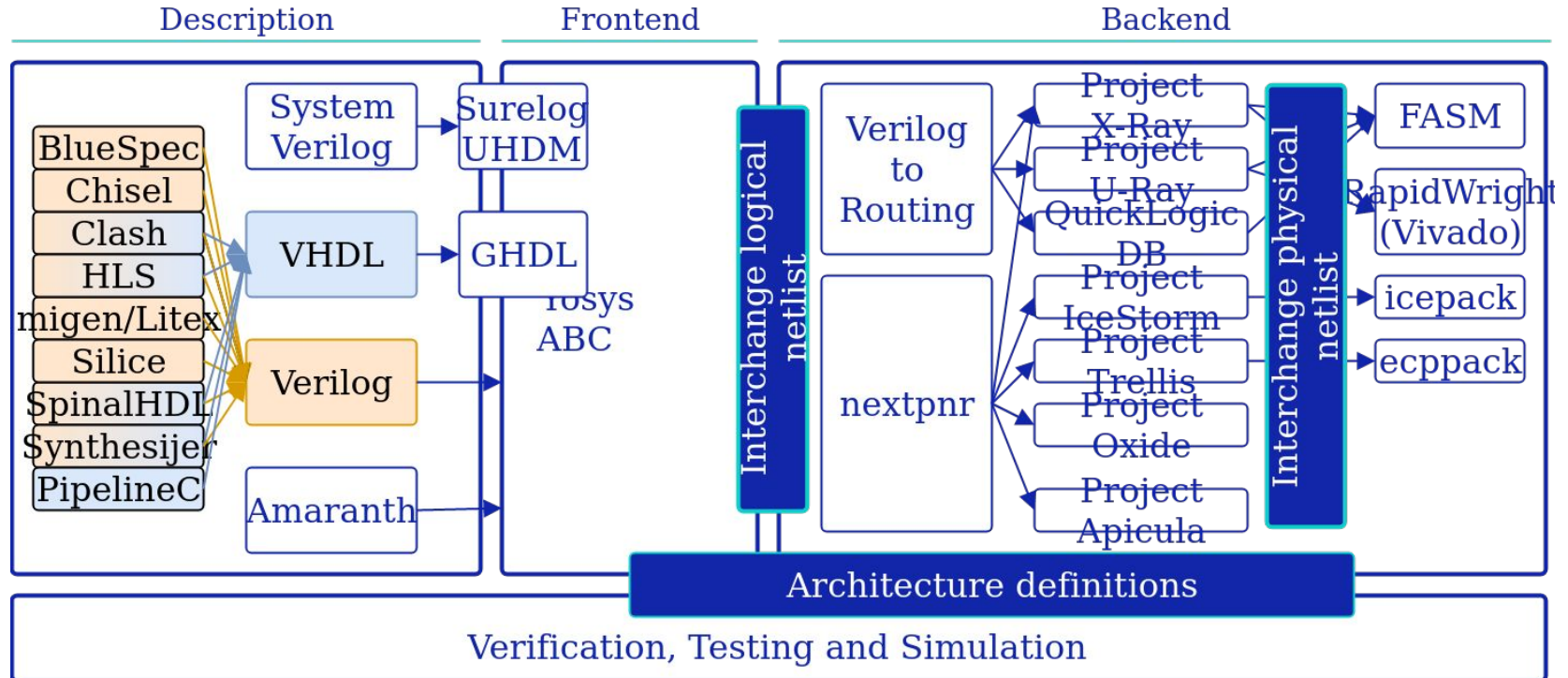
- Antes ya existían herramientas open-source, sobre todo, para simulación: Icarus Verilog, Verilator o GTKWave, por ejemplo.
- Proyecto compuesto por diferentes partes que cubre el workflow de diseño de FPGAs.
- Mucha documentación y ingeniería inversa para entender los formatos de bistreams de cada fabricante
- Cantidad limitada de FPGAs soportadas.
- Un futuro prometedor!



# F4PGA - Arquitectura



# F4PGA - Arquitectura



# F4PGA - FPGA soportadas

	Project Icestorm	Project Trellis	Project X-Ray	QuickLogic Database
<b>Basic Tiles:</b> <ul style="list-style-type: none"> <li>- Logic</li> <li>- Block RAM</li> </ul>	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✗	✓ ✓ ✓
<b>Advanced Tiles:</b> <ul style="list-style-type: none"> <li>- DSP</li> <li>- Hard Blocks</li> <li>- Clock Tiles</li> <li>- IO Tiles</li> </ul>	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓	✓ <sub>x</sub> ✗ ✗ ✓	✓ ✓ ✓ ✓
<b>Routing:</b> <ul style="list-style-type: none"> <li>- Logic</li> <li>- Clock</li> </ul>	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓

**Xilinx 7-Series**

The most popular Xilinx FPGA family

[learn more](#)

**Lattice ice40**

World's smallest FPGAs for mobile devices

[learn more](#)

**Lattice ecp5**

Low cost FPGAs with high performance features

[learn more](#)

**QuickLogic EOS S3**

FPGA + CPU sensor processing platform

[learn more](#)

**QuickLogic QLF K4N8**

A 24x24 eFPGA with 6144 flip-flops, 4608 LUT4s, adder and shift-register support

[learn more](#)

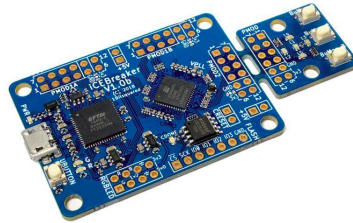
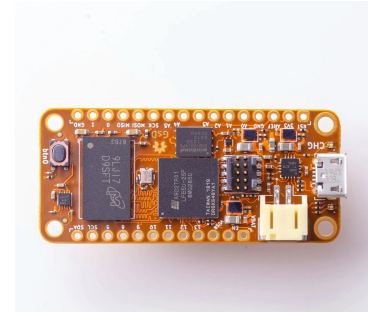
**Do you want to add more?**

Help us!

[learn more](#)

# Ecosistema - placas Open-Hardware

- Icezum [Alhambra](#)
- [ICEBreaker](#)
- [OrangeCrab](#) (Linux ✓)
- [ULX3S](#) (Linux ✓)



# Ecosistema - herramientas

## ● SoftCores

- [LibreCores](#)
- [OpenCores](#)
- Puedes encontrar muchas IP Cores listas para utilizar
- De todo tipo, comunicaciones, controladores, ...

## ● Simuladores

- Permiten verificar la logica codificada
- [Icarus Verilog](#)
- [Verilator](#)

Processor 218

Project	Files	Statistics	Status	License	Wishbone version
<a href="#">16 Bit Microcontroller</a>	●	Stats	<a href="#">[wbc]</a>	LGPL	
<a href="#">16-bit CPU based loosely on Caxton Foster's Blue architecture</a>	●	Stats		LGPL	
<a href="#">16-bit Open uRISC core Processor</a>	●	Stats		LGPL	
<a href="#">1664 microprocessor</a>	●	Stats		Others	
<a href="#">32 bit Processor</a>	●	Stats		LGPL	
<a href="#">4004 CPU and MCS-4 family chips</a>	●	Stats		Others	
<a href="#">6502VHDL</a>	●	Stats			
<a href="#">6809 and 6309 Compatible core</a>	●	Stats		LGPL	
<a href="#">68hc05</a>	●	Stats			
<a href="#">68hc08</a>	●	Stats			
<a href="#">8-BIT HARDWIRED PROCESSOR</a>	●	Stats	<a href="#">[done]</a>	LGPL	
<a href="#">8-bit microcontroller with extended peripheral set</a>	●	Stats		LGPL	
<a href="#">8-BIT MICROPROGRAMMED PROCESSOR</a>	●	Stats	<a href="#">[done]</a>	LGPL	
<a href="#">8-bit Pipelined Processor</a>	●	Stats		LGPL	
<a href="#">8-bit uP</a>	●	Stats	<a href="#">[done]</a>	GPL	
<a href="#">8051 core</a>	●	Stats	<a href="#">[wbc]</a>		
<a href="#">8080 Compatible CPU</a>	●	Stats	<a href="#">[done]</a>		
<a href="#">A lightweight 8085 verilog implementation</a>	■	Stats		LGPL	
<a href="#">A-Z80 CPU</a>	●	Stats	<a href="#">[done]</a>	LGPL	
<a href="#">ae18</a>	●	Stats	<a href="#">[done]</a> <a href="#">[wbc]</a>	LGPL	
<a href="#">aeMB</a>	●	Stats	<a href="#">[done]</a> <a href="#">[wbc]</a>	LGPL	
<a href="#">ag_6502 soft core with phase-level accuracy</a>	●	Stats	<a href="#">[done]</a>	GPL	
<a href="#">AltQ32 - Alternative Lightweight OpenRisc CPU</a>	●	Stats	<a href="#">[done]</a> <a href="#">[wbc]</a>	LGPL	
<a href="#">alt_1SA</a>	●	Stats		LGPL	
<a href="#">Alwcpu - A light weight CPU</a>	●	Stats	<a href="#">[done]</a> <a href="#">[wbc]</a>	LGPL	

# Ecosistema - algunas herramientas

- [FuseSoc](#)
  - Es un package manager de IP Cores
  - Contiene sistema para compilar los cores
  - El objetivo es maximizar la reutilización de los cores

# Ecosistema - algunas herramientas

- RISC-V
  - Existen cores totalmente open-source, donde se siguen [estándares](#) de RISC-V
  - Desde microcontroladores de 8-bits, a microprocesadores de 32-bits capaces de ejecutar Linux.
- Ejemplos:
  - [Picorv32](#)
  - Picorv32 en Alhambra, [aquí](#)
  - [Serv](#), microcontrolador de 8 bits

# Ecosistema - algunas herramientas

- LiteX
  - Herramienta basada en Python para generación de SoCs completos
  - Utiliza Mygen (lenguaje HDL) para codificar.
  - Pueden contener
    - CPUs
    - Controladores, PCIe, memoria SRAM, ...
    - Perifericos, I2C, SPI, Ethernet, HDMI,...
  - Presentacion, [aquí](#)

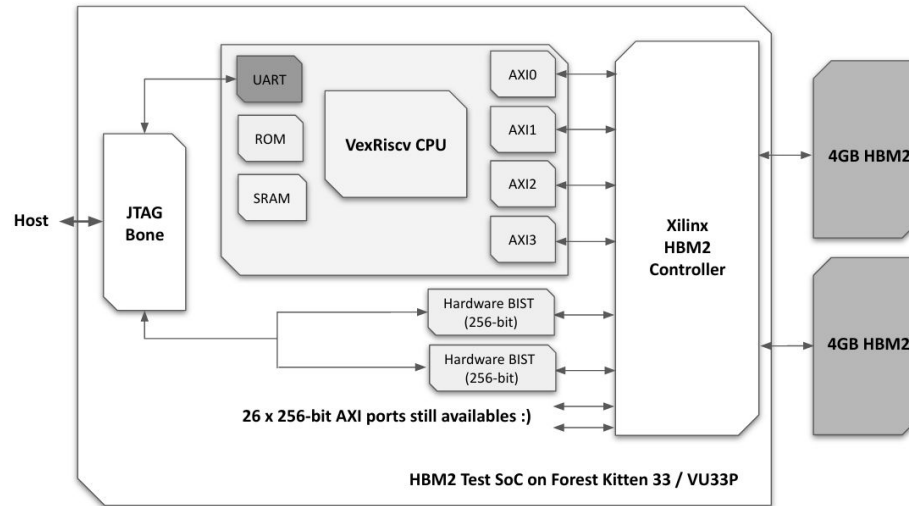


# Ecosistema - algunas herramientas

- Existen nuevos lenguajes de programación de hdl: [Chisel](#), nmigen o [spinalhdl](#), mas modernos.

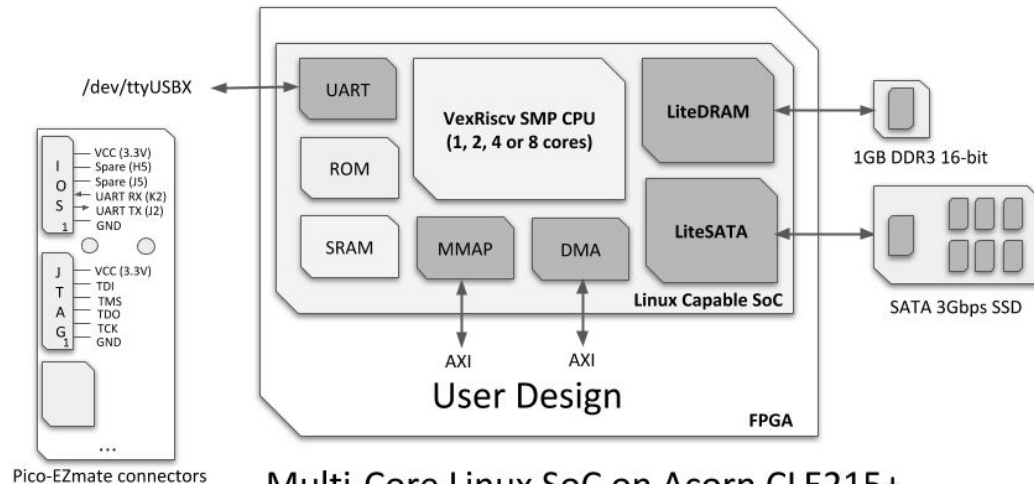
# Ecosistema - algunas herramientas

- LiteX - Ejemplos



# Ecosistema - algunas herramientas

- LiteX - Ejemplos



Multi-Core Linux SoC on Acorn CLE215+

EnjoyDigital  
CD

Built with



# Demo

- [Blinky](#) en Alhambra y ULX3S
- [UART](#) en Alhambra
- ~~PicoRV en Alhambra~~
- ~~LiteX~~

# Links Adicionales

- Tutoriales Verilog/Vhdl:
  - <https://github.com/Obijuan/open-fpga-verilog-tutorial>
  - <https://www.nandland.com/>
  - [https://www.reddit.com/r/FPGA/comments/omrnrk/list\\_of\\_useful\\_links\\_for\\_beginners\\_and\\_veterans/](https://www.reddit.com/r/FPGA/comments/omrnrk/list_of_useful_links_for_beginners_and_veterans/)
  - <https://www.fpga4fun.com/SiteInformation.html>

Eskerrik asko zuen arregatik!

Galderarik?