

FPGAs Libres: Compartiendo Hardware



Juan González Gómez
@Obijuan_cube
<https://github.com/Obijuan>



Universidad
Rey Juan Carlos



Universidad
Rey Juan Carlos

Oficina de
Conocimiento y
Cultura Libres

*Día del Software Libre
Campus de Móstoles, URJC
23-Sep-2019*



¡Hola! Como esta es una charla corta, voy a ir directo al grano. El mensaje más importante es: Por primera vez en 30 años disponemos de **herramientas Libres** para cerrar el **ciclo completo de diseño de hardware** en FPGA, desde el circuito fuente hasta la generación del **Bitstream** y su carga en la FPGA. Ahora veremos qué significa todo esto. Pero esto es algo que va a tener un impacto muy fuerte a medio largo plazo en la informática y la ingeniería

4 Ideas

Compartir es muy potente

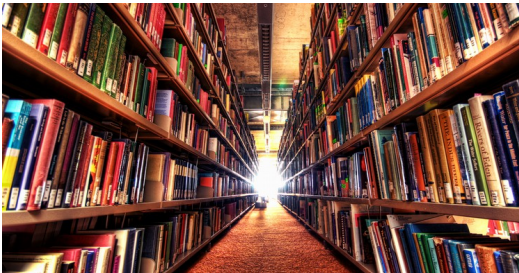
Hardware Líquido

Herramientas de desarrollo Libres

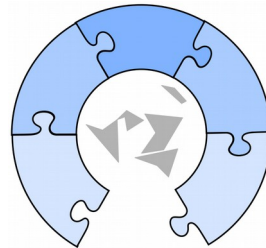
FPGAs Libres

Me voy a centrar en **4 ideas claves**

Compartir es muy potente



***Bibliotecas de
conocimientos tecnológico***



***Patrimonio tecnológico
de la humanidad***

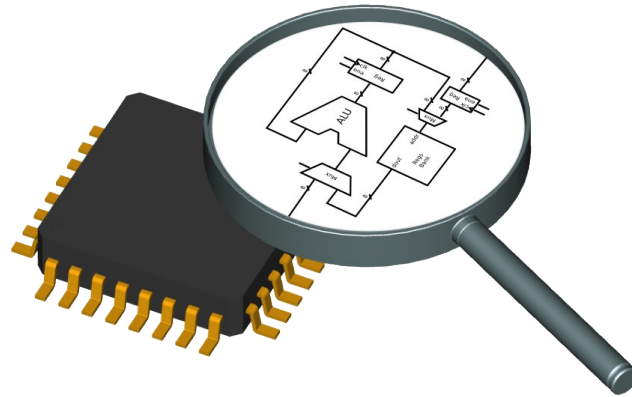
Primera idea: Compartir es muy potente

Una de las grandes lecciones que nos ha enseñado el software libre en esto casi 40 años es que **compartir el código fuente funciona**. Al compartirlo hemos creado entre todos millones de líneas de código que hacen que miles de servidores, máquinas, internet, tablets, móviles en todo el mundo funcionen. Es una **gran biblioteca de conocimiento acumulado**, que a mí me gusta llamarlo **Patrimonio tecnológico de la Humanidad**. Está ahí para que todos lo podamos usar, lo podamos estudiar, lo podamos mejorar. Las empresas van y viene, pero lo libre permanece. Nuestro patrimonio permanece

**Imaginaros qué podría ocurrir
si pudiésemos compartir el
hardware de igual manera que
el software**

Imaginaros que pudiésemos **compartir el hardware** de la misma manera que hacemos con el software. Me imagino un ordenador con el hardware “en blanco” donde te puedes **descargar el procesador** que quieras, y le puedas añadir hardware específico según lo que necesites: aceleración gráfica para trabajar en 3D, unidades de criptografía, o lo que sea. Me imagino **repositorios** con controladores hardware, donde hay una comunidad desarrollándolos. Te los bajas, los pruebas, los modificas, los subes... Podríamos incluso crear **distribuciones de hardware**, con todos los componentes ya listos para usar **¿Cómo lo podemos hacer posible?**

Hardware Líquido

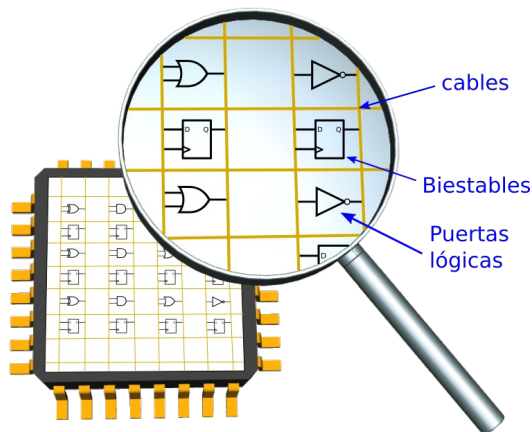


Cualquier circuito digital, por muy complejo que sea, se descompone en estos 3 tipos de **componentes elementales**: Puertas lógicas, cables y biestables

Esto nos lleva a la segunda idea clave: el **hardware líquido**, o hardware blando, o soft-hardware

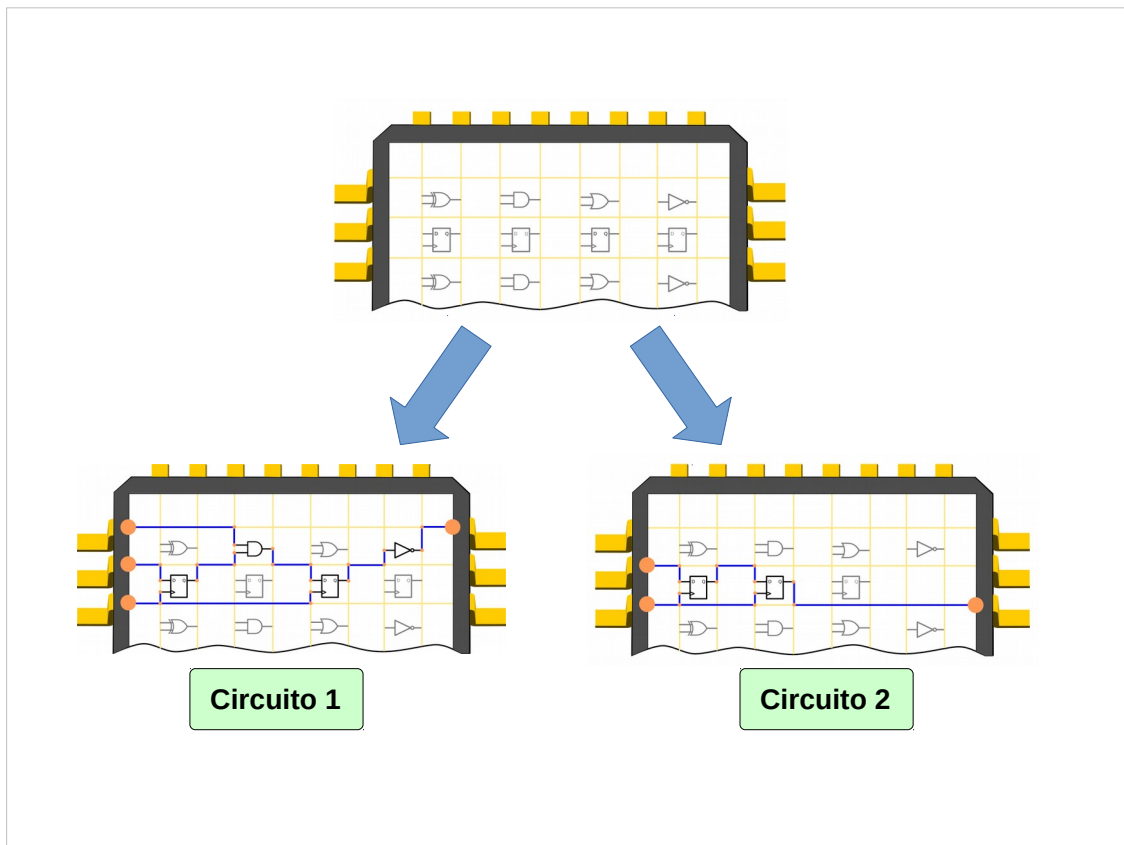
Cualquier **circuito digital**, por muy avanzado que sea, está formado por sólo **3 componentes básicos**: Las **puertas lógica** para manipular los bits y combinarlos, los **cables** para transportar los bits, y los **biestables** para almacenarlo

Tecnología FPGA



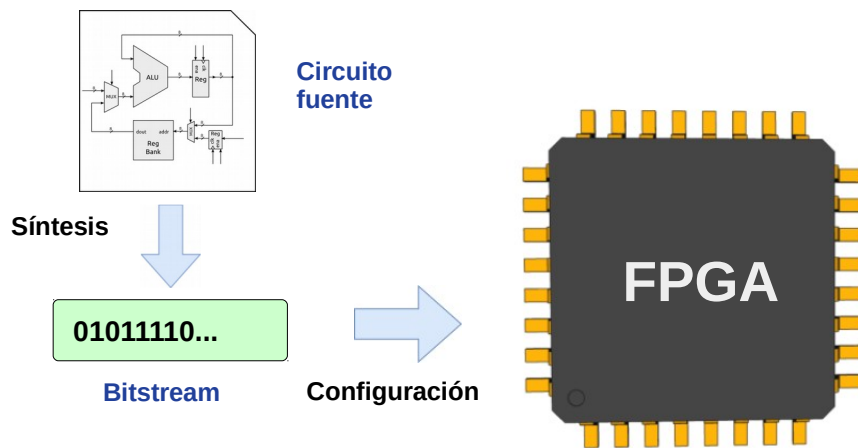
FPGA: Chip “en blanco” que contiene una matriz con los 3 componentes básicos: puertas lógicas, biestables y cables

Desde los años 80, curiosamente los años del nacimiento del software libre, existen unos chips llamados **FPGAs** que en su interior tienen estos tres componentes: puertas lógicas, biestables y cables. Pero las uniones entre ellos NO están establecidas. Estas uniones son **configurables**. Si colocamos un bit a 1 se establece la unión entre dos elementos, y si lo dejamos a 0, NO



De esta forma **establecemos** las uniones necesarias entre los componentes para crear nuestro **circuito digital**. Partimos de una FPGA sin configurar, establecemos unas uniones y obtenemos el **Circuito 1** (que podría ser un procesador por ejemplo). Podemos volver al estado inicial, quitar estas uniones y crear otras nuevas, con lo que obtendríamos otro circuito digital diferente: el circuito 2. A este proceso de establecer las uniones lo llamamos **configuración de la FPGA**

Compartiendo Hardware



El **BITSTREAM** son los bits de configuración para establecer las uniones entre los elementos lógicos. ¡Es lo que nos permite compartir hardware!

Al conjunto de todos los **bits de configuración**, que determina qué uniones habrá y por tanto qué hardware será **sintetizado**, lo llamamos el **Bitstream**: el “chorro” de bits

Al cargar este bitstream en la FPGA, “aparece” nuestro hardware. Es este bitstream el que compartimos para descargamos el hardware. Este bitstream lo generamos a partir de nuestro circuito fuente usando las **herramientas de desarrollo**

Herramientas de Desarrollo Libres



- **GNU GCC:** Primer compilador libre de la historia

- **Autonomía**

Y llegamos a la tercera idea clave: **Herramientas de desarrollo Libre**

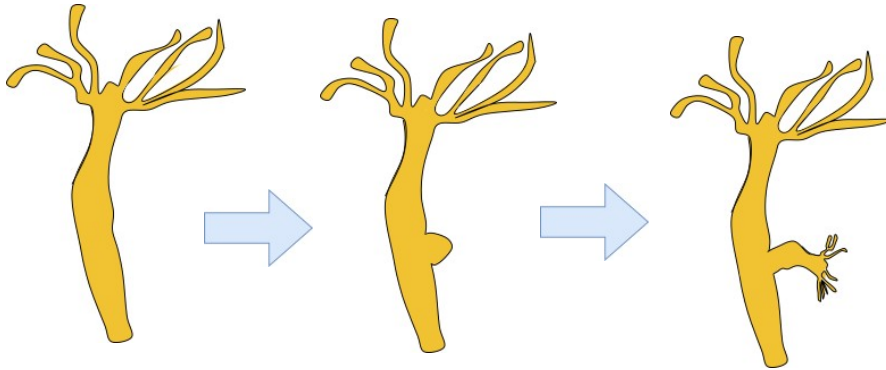
En el inicio del software libre, a principios de los 80, hubo un hito muy importante: La creación del compilador **GNU/GCC**: El primer **compilador libre** de la historia. Fue un hito fundamental porque a partir de ahí ya se podía crear software libre, pero usando sólo herramientas libres. O dicho de otra forma, poder crear aplicaciones libres desde dentro del ecosistema libre. Este ecosistema libre se convirtió en algo autocontenido, y autónomo, eliminando las dependencias con los ecosistemas anteriores

Origen del primer ecosistema libre (I)

Ecosistema original:
Herramientas privativas

Desarrollo de
tu aplicación

Tu aplicación crece, pero
está atada al ecosistema
original



Esto es la clave. Me gusta visualizarlo gráficamente usando un símil de la biología: el proceso de gemación de las Hidras. Son unos bichos invertebrados que se reproducen asexualmente mediante gemación.

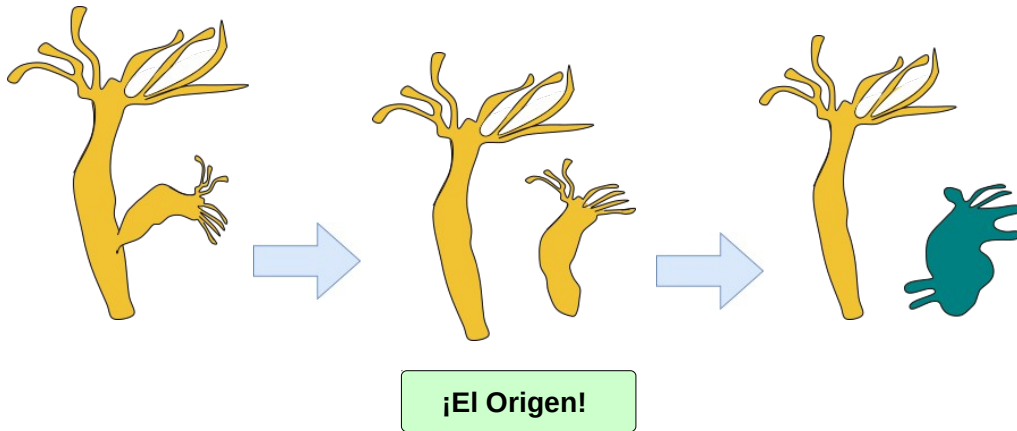
La hydra padre es el ecosistema inicial, donde el software es privativo. Los primeros programas libres se crearon dentro de este ecosistema, con compiladores privativos. Si bien son programas libres, están “ATADOS” al ecosistema padre. La empresa creadora del compilador usado puede decirnos quién lo puede usar o quién no. O en qué países sí y qué otros no. O en qué plataformas sí y en qué no. Hay una restricción muy fuerte que hace que compartir sea complicado. Se pierde esa potencia de compartir

Origen del primer ecosistema libre (II)

Pero... si consigues
Herramientas libres...

Tu aplicación se convierte
en independiente

Ahora puede evolucionar
como la comunidad decida

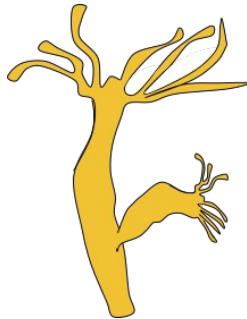


Por eso, desde el principio, la FSF se centró en hacer primero las **herramientas de desarrollo libres**, y completar así ese proceso de gemación en que aparece un **ecosistema nuevo**, independiente del padre. Si ahora el padre muere, nos da igual. Nosotros seguimos vivos. No nos pueden poner restricciones. Este es el verdadero origen del software libre: el nombre en el que nace un ecosistema independiente. A partir de aquí ya puedes evolucionar tu ecosistema.

Y esto sólo es posible por tener **herramientas de desarrollo libres**

FPGAs Libres

Ecosistema FPGAs
tradicionales



Nuestros
circuitos

¡El Fabricante no proporciona
la información del Bitstream!

Durante 30 años esta ha sido la situación. Sólo puedes usar las herramientas proporcionadas por el Fabricante, y mediante los términos que ellos han establecido

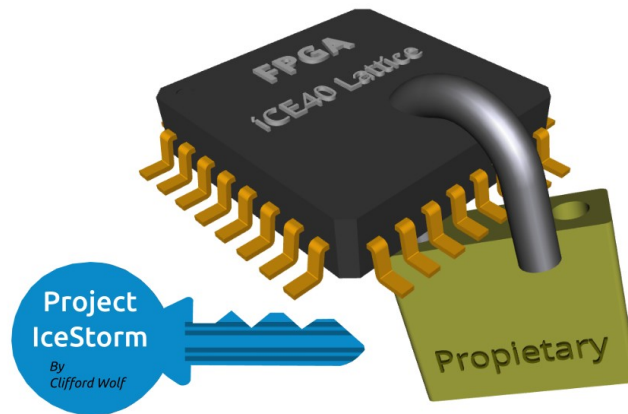
Las comunidades y todo lo que han generado están ATADAS a un fabricante específico

Y llegamos a la cuarta y última idea Clave: Las FPGAs Libres.

Este es el escenario de las FPGAs tradicionales. El proceso de Generación no se ha completado. Nuestros diseños están atados a las herramientas de desarrollo del fabricante. No está disponible la información sobre el bitstream. No lo podemos generar. Sólo con las herramientas del fabricante. Estamos Atados.

Aunque las FPGAs se iniciaron en los 80, este ecosistema ha frenado la compartición de hardware. Si bien han aparecido comunidades y se han hecho diseños libres, debido a estas ataduras con el fabricante no se ha conseguido llegar ni de lejos a donde ha llegado el software libre. La compartición es muy difícil. Se pierde toda esa potencia de compartir

FPGAs libres: El renacimiento

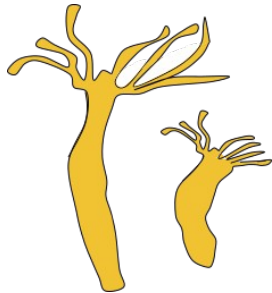


- Proyecto Icestorm (Mayo, 2015)
- La primera *toolchain* que permiten pasar de Verilog al bitstream usando sólo Herramientas libres

PERO, en mayo del 2015, apareció el **primer sintetizador libre de la historia**, creado por Clifford Wolf, un ingeniero Austriaco. Tardó 3 años. Hizo ingeniería inversa de una familia de FPGAs de Lattice, la Ice40, y logró obtener todos los detalles, que documentó y publicó: proyecto icesstorm. Y se creó la primera comunidad libre y se pudo, por primera vez, empezar a compartir y aprovechar de esta potencia de compartir

Conclusiones

Ecosistema FPGAs
Privativas



¡Estamos en
el Origen!

Ecosistema
FPGAs Libres



- ¡Ya podemos **crear y compartir hardware** de manera autónoma!
- ¡Está todo por hacer!
- Hay que re-hacer el **Hardware**, pero desde el ecosistema **LIBRE**

Gracias a este hito se completó el proceso de Gemación donde apareció un ecosistema independiente del padre. Donde se pudo crear hardware en FPGAs con independencia del fabricante. Donde ya no se aplican las restricciones que tienen en sus herramientas. Estamos exactamente igual que como estaba el software libre en el nacimiento del GCC, pero casi 40 años después

Ahora ya podemos compartir el hardware de manera autónoma. Es el momento de re-hacer todos los circuitos digitales, pero con estas herramientas libres, de igual forma que la FSF en el proyecto GNU re-hicieron todas las herramientas, pero libres

Es una labor titánica, pero apasionante. Así que yo os invito a todos a que os unaís y os embarquéis en este viaje

**¡Que las FPGAs libres os
acompañen!**



FPGAs Libres: Compartiendo Hardware



Juan González Gómez
[@Objuan_cube](https://github.com/Objuan)
<https://github.com/Objuan>



Universidad
Rey Juan Carlos

Oficina de
Conocimiento y
Cultura Libres

*Día del Software Libre
Campus de Móstoles, URJC
23-Sep-2019*

