



Instituto  
Nacional  
de Tecnología  
Industrial

**SASE 2011**  
Simposio Argentino de Sistemas Embebidos

Ministerio de Industria  
Secretaría de industria y Comercio



## Desarrollo con FPGAs en GNU/Linux

Autores:

**Ing. Salvador E. Tropea**

**Ing. Rodrigo A. Melo**

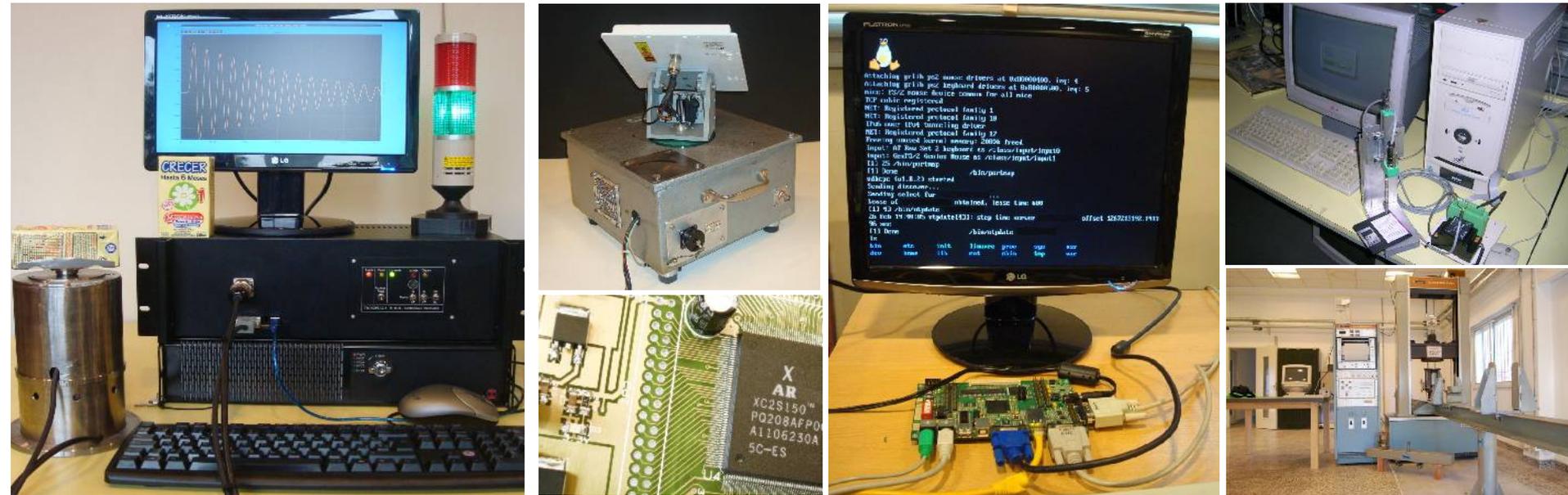
**Ing. Diego J. Brengi**

Electrónica e Informática

Unidad Técnica Instrumentación y Control

Desarrollo Electrónico con Software Libre (DESoL)





## Desarrollo con FPGAs en GNU/Linux

Desarrollo con dispositivos FPGA utilizando un entorno GNU/Linux y herramientas de software libre.

## Agenda General

1. Presentación INTI y Laboratorio DESoL
2. Motivos para usar GNU/Linux (Software Libre)
3. El ciclo de trabajo con FPGAs
4. Herramientas de software más relevantes
5. Demostración y consultas

## Agenda

# Presentación

- **Instituto Nacional de Tecnología Industrial**
- **Centro de electrónica e Informática**
- **Laboratorio de Desarrollo Electrónico con Software Libre**
- **Áreas de trabajo del laboratorio**

## Instituto Nacional de Tecnología Industrial

El Instituto Nacional de Tecnología Industrial (INTI) es una institución nacional creada en 1957 para promover el desarrollo y la transferencia de tecnología a la industria.

### Misión del INTI

- Responsable técnico en la aplicación de las regulaciones oficiales de calidad o identidad de productos en la industria.
- Asistente público para la competitividad de empresas industriales o de servicios industriales y de los sectores que las agrupan, en todo el país.
- Responsable tecnológico público de procurar la integración al tejido productivo de toda la comunidad, en todo el país, en los aspectos industriales y vinculados.

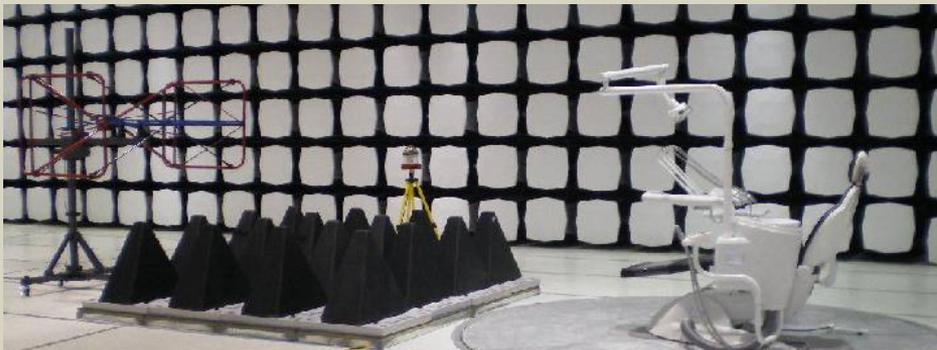
<http://www.inti.gov.ar/pdf/internol.pdf>



## INTI - Centro de Electrónica e Informática

El centro de Electrónica e Informática tiene como principal objetivo apoyar el desarrollo tecnológico del subsector industrial relacionado, a través de desarrollos precompetitivos, asistencia técnica, ensayos, calibraciones y certificaciones, enmarcado en el Plan Estratégico del INTI.

<http://www.inti.gov.ar/electronicaeinformatica/>



*Cámara semianecoica electromagnética*



*Banco de mezcla de gases*



*Sala limpia*



*Medición de un transmisor de TV digital isdb-t*

## Laboratorio de Desarrollo Electrónico con Software Libre



Av. Gral. Paz 5445  
(Constituyentes y Albarellos)  
CC 157 - (CP 1650)  
Edificio 42- San Martín  
**Provincia de Buenos Aires**  
**República Argentina**

**Instituto Nacional de Tecnología Industrial**

**Centro de Electrónica e Informática**

**Unidad Técnica de Instrumentación y Control**

**Laboratorio de Desarrollo Electrónico con Software Libre**

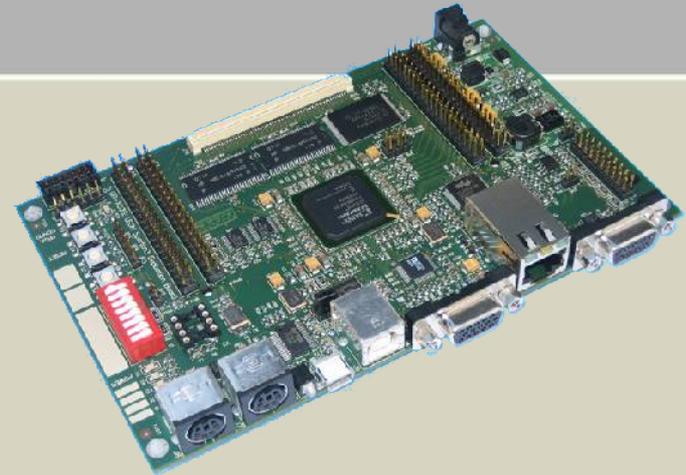
## Áreas de Trabajo - Parte 1/3

### Aplicación de dispositivos lógicos programables FPGA (Field Programmable Gate Array).

Utilización de dispositivos FPGA en aplicaciones que requieran de alta velocidad, gran flexibilidad o soluciones no convencionales.

### Diseño y adaptación de IP cores en lenguaje VHDL portable.

- Diseño de IP cores (bloques reutilizables para FPGAs y ASICs).
- Utilizando lenguaje VHDL portable para permitir su utilización en casi cualquier dispositivo FPGA (y/o ASIC).



*Sistema FPGA capaz de correr GNU/Linux*

```
Archivo Editor Buscar Macro Rec.16 83
[*]..cores/epp_wb/epp_wb_top.vhdl[0]
library epp2wb;
use epp2wb.Devices.all;
library IEEE;
use IEEE.std_logic_1164.all;

entity Fpp_wb_top is
  generic(
    FTI_TFR_DEPTH      : positive:=1;
    FILTER_DATA        : boolean:=fa
  );
  port(
    Parallel port signals
    brd_lpt_nack_o      : out  std_l
    brd_lpt_busy_o     : out  std_l
    brd_lpt_nerr_o     : out  std_l
    brd_lpt_pe_o       : out  std_l
    brd_lpt_sel_o      : out  std_l
  );
end entity;
136:2
```

*Editor adaptado para código VHDL*

## Áreas de Trabajo - Parte 2/3

**Equipos dedicados utilizando GNU/Linux y software libre.**

**Integración de sistemas a medida utilizando estándares abiertos.**

Desarrollo y modernización de sistemas dedicados y aplicaciones embebidas especiales utilizando PC para las áreas de control, ensayos, monitoreo y registro. Aprovechando protocolos y estándares abiertos con sistema operativo GNU/Linux y aplicaciones de software libre.

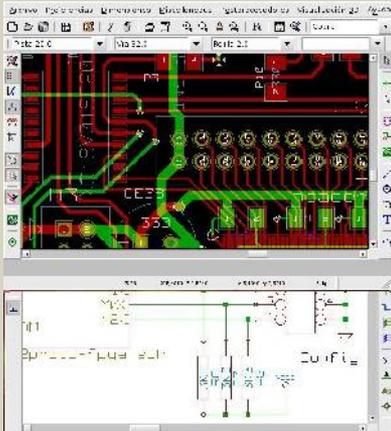


*Modernización con GNU/Linux de un equipo de ensayos para maderas*

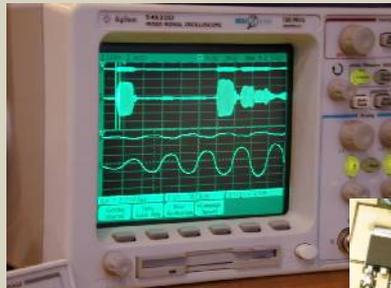


*Sistema de control de calidad para lácteos.*

## Áreas de Trabajo - Parte 3/3



*Software libre para diseño de PCB*



*Medición de sensores  
ultrasónicos*

*Diseño de Hardware  
libre con FPGA*



**Herramientas de software libre aplicadas al desarrollo electrónico.**

**Instrumentación y control electrónico utilizando microcontroladores, sensores y actuadores.**

**Diseño conjunto de hardware y software.**

## Agenda

# Motivos para Usar GNU/Linux en el Laboratorio

- **El Software Libre**
- **Sistemas GNU/Linux**
- **Ventajas del Software libre**

## El Software Libre

**Es el software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente.**

Formalmente se lo define como el que garantiza las siguientes libertades:

- **Libertad 0:** usar el programa, con cualquier propósito.
- **Libertad 1:** estudiar cómo funciona el programa, y adaptarlo a tus necesidades.
- **Libertad 2:** distribuir copias, con lo que puedes ayudar a tu vecino.
- **Libertad 3:** mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

Las libertades 1 y 3 requieren que esté disponible el código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.

*Es importante aclarar la diferencia entre software gratuito y el libre ya que mucha gente confunde estos conceptos.*

**El concepto y la definición de “Software Libre” son establecidos por la Free Software Foundation: <http://www.fsf.org/>**

## Sistemas GNU/Linux

**Es la combinación de dos componentes principales:**

- El núcleo (kernel) llamado Linux.
- Herramientas del proyecto GNU (de la Free Software Foundation).



Estas dos componentes forman un sistema operativo completo que cumple con las libertades mencionadas del Software Libre.

Además, un Sistema GNU/Linux incorpora en la actualidad una gran cantidad de software libre adicional (herramientas de oficina, aplicaciones científicas, de administración y muchas más).

Las diferentes variantes de estos sistemas se llaman distribuciones. Las distribuciones pueden incorporar paquetes de software no-libre.

## Ventajas técnicas

- **Amplia disponibilidad de recursos**
  - Lenguajes de programación y herramientas.
  - Bibliotecas, rutinas, etc.
  - Comunidad de usuarios predispuestos.
- **Menor esfuerzo de mantenimiento y administración**
  - Administración más simple y centralizada.
  - Manejo coherente de paquetes.
  - Sistemas seguros y muy estables. No se degradan con el uso.
  - Muy apto para sistemas remotos.
- **Mayor control**
  - Código fuente disponible (adaptable y/o corregible)
  - Componentes altamente configurables.
  - Acceso a todos los protocolos de comunicación y formatos de archivos.
  - Conocimiento del hardware/firmware involucrado (Hardware libre o abierto).
- **Muchas de las herramientas superan a sus equivalentes comerciales** (otras necesitan de adaptación y desarrollo adicional).



## Ventajas estratégicas

- **Autonomía nacional**
  - Reduce la dependencia de corporaciones extranjeras
  - Fomenta el desarrollo local de software y hardware.
- **El conocimiento más profundo de las herramientas de software para electrónica (propias y de terceros) brinda normalmente mayor experticia en la temática ya que hoy en día el desarrollo de hardware está muy ligado al software.**



## Ventajas económicas

- **Sin costos de licencias de software**
  - Para el desarrollador.
  - Para el cliente.
    - OS y herramientas incluidas en el equipo
    - Bibliotecas
    - Herramientas para reproducir el desarrollo (ej: KICAD)
- **El dinero puede utilizarse o aprovecharse en hardware.**
- **Fácil aplicación en sistemas de pocos recursos de hardware.**

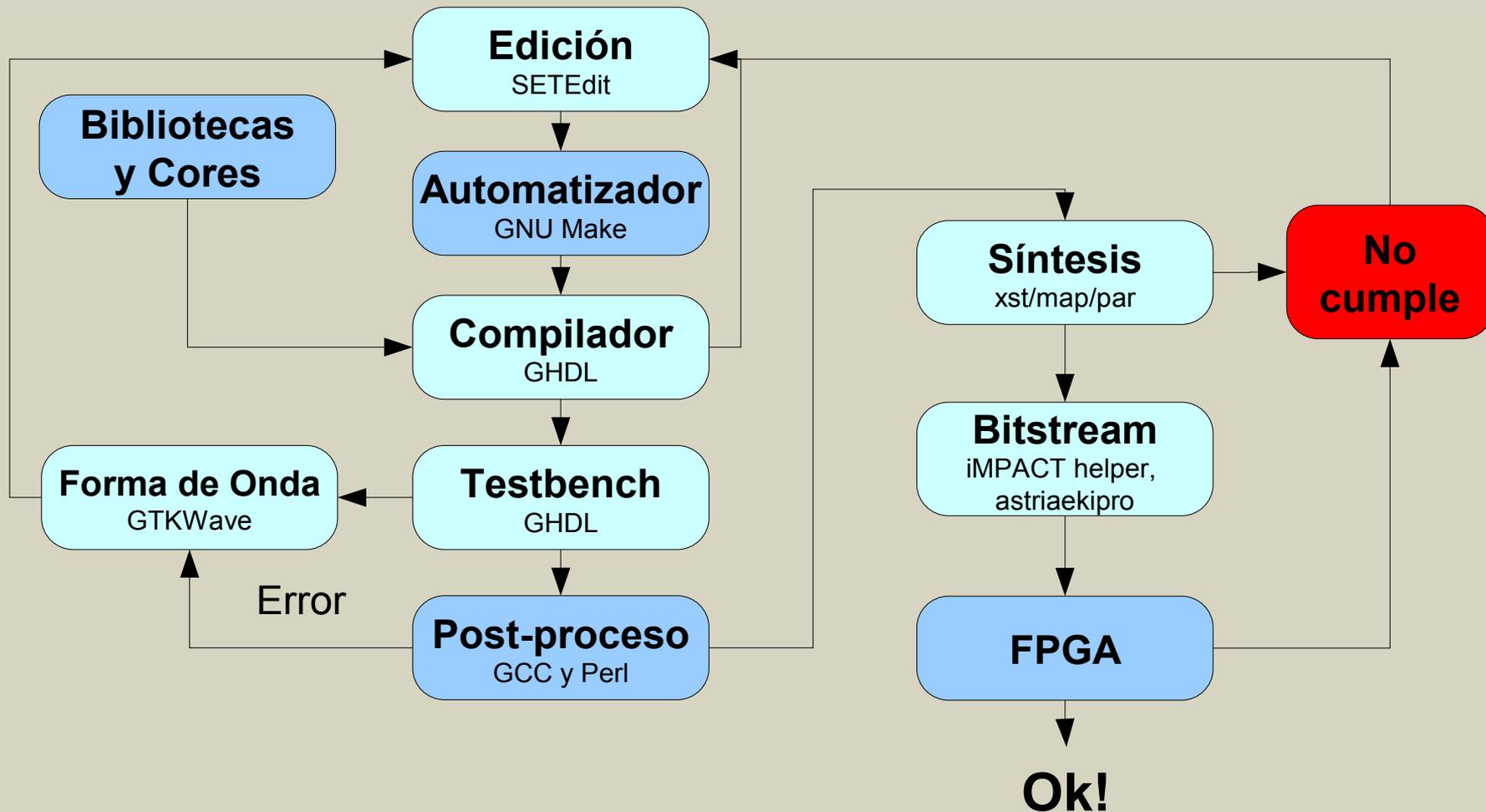


## Agenda

# El Ciclo de Trabajo con FPGAs

- **Diagrama en bloques simplificado**

## Ciclo básico de desarrollo



## Agenda

# Herramientas de Software más Relevantes

- **Sistema Operativo**
- **Herramientas de propósitos generales**
- **Edición: Setedit**
- **Automatización: Gnu Make**
- **Simulación y testbench: GHDL**
- **Análisis: GTKWave**
- **ISE Webpack**
- **Transferencia del bitstream**
- **Ayudas: Xil Project**
- **Hardware: Placa de desarrollo con FPGA**
- **Circuitos Impresos: Kicad**
- **Reportes y Documentación**

## Sistema operativo Debian GNU/Linux

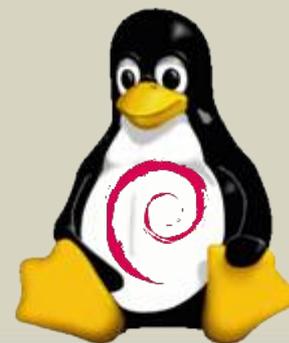
- Sistema operativo: Debian GNU/Linux Estable
- Gran cantidad de paquetes de software listos para usar. Squeeze: 29.000 paquetes aprox. (52 CDs u 8 DVDs!).
- Más de 3000 voluntarios contribuyen a su mejora y desarrollo.
- No posee dependencia directa de compañías o empresas.
- Es uno de los proyectos Open Source (y de software en general) más grandes del mundo.
- Una de las distribuciones GNU/Linux más antiguas que sigue activa.
- Ubuntu es una distribución basada en Debian.



+



=



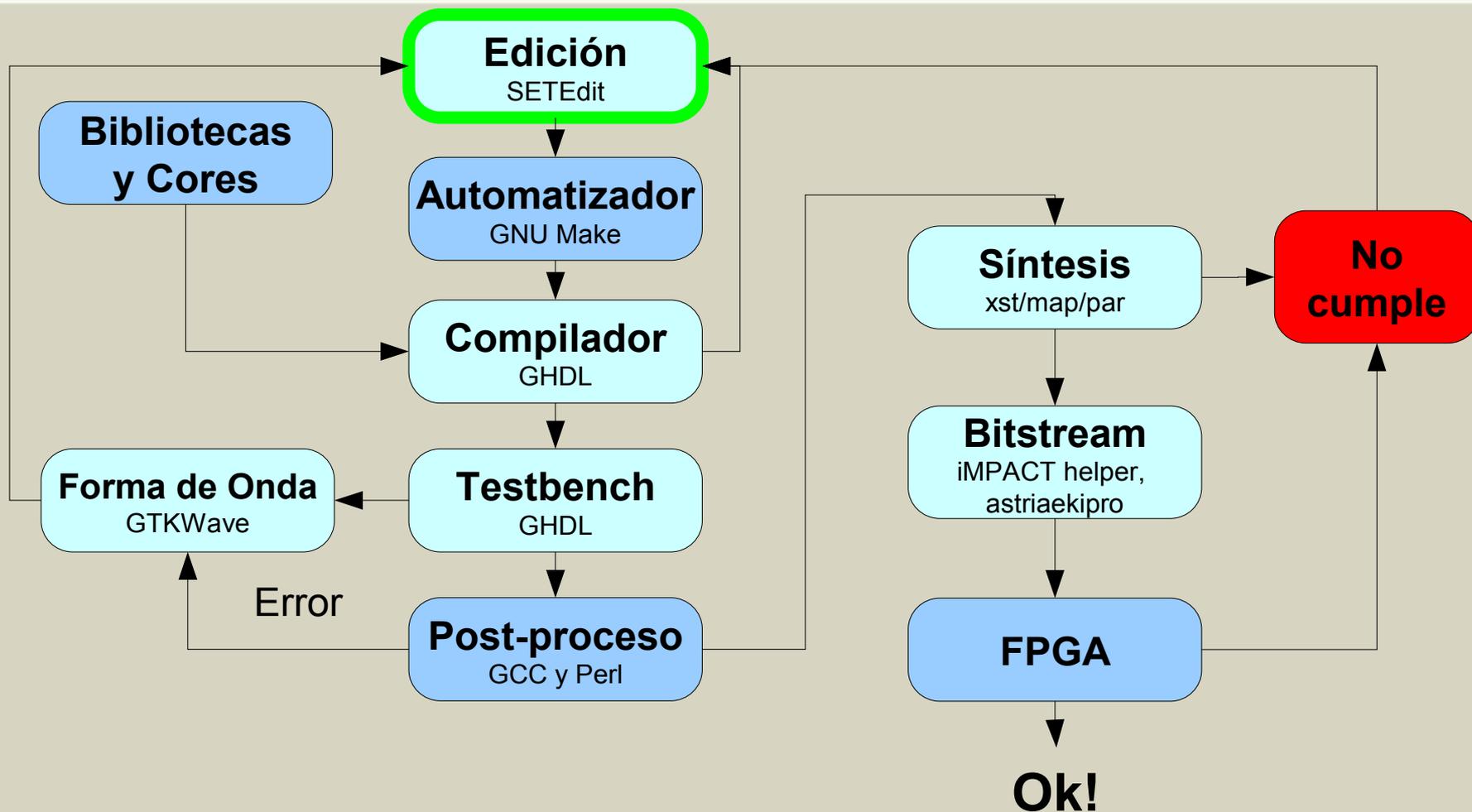
## Herramientas de propósitos generales

- Sistema Operativo: Debian GNU/Linux versión estable
- Suite de oficina: OpenOffice Writer – Calc – Impress – Draw
- Navegador Web: IceWeasel (Firefox)
- Cliente de Correo: IceDove (ThunderBird)
- Consola: Eterm
- Entornos de escritorio:
  - Gnome (máquinas modernas)
  - Enlightenment
- Navegación de sistema de archivos:
  - GNU Midnight Commander (consola)
  - Nautilus (gráfico)
- Visor de imágenes: gqview
- Edición de imágenes: Gimp
- Visor de archivos pdf: xpdf
- Acceso remoto entre terminales: OpenSSH
- Cálculos y gráficos: octave y gnuplot



**GNUPLLOT**

## Edición: SETEdit

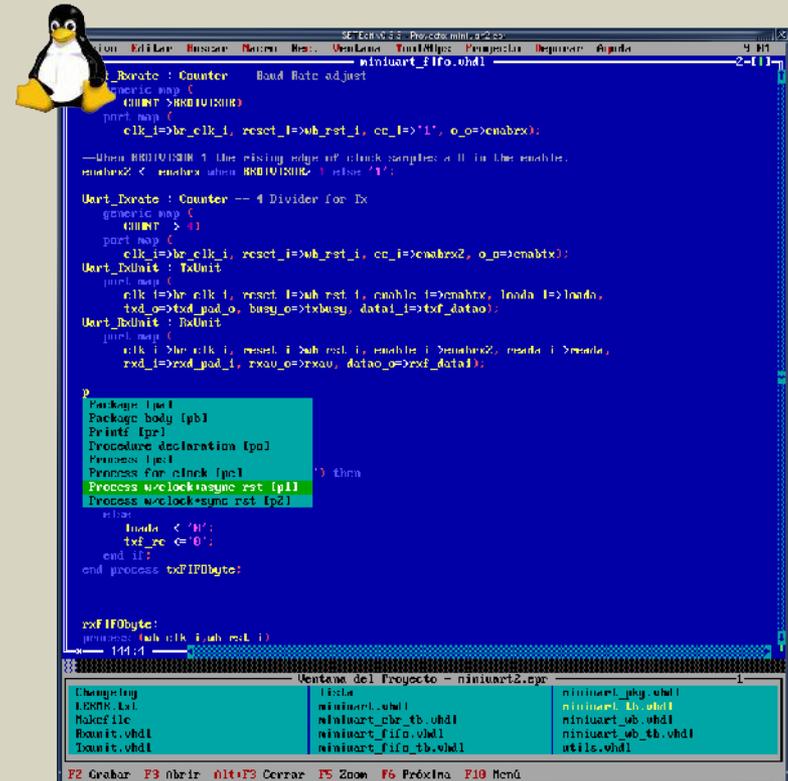


## Edición: SETEdit

Editor de software libre que posee facilidades avanzadas para la edición de código VHDL. Ha sido desarrollado para programadores y posee soporte para gran cantidad de lenguajes de programación.

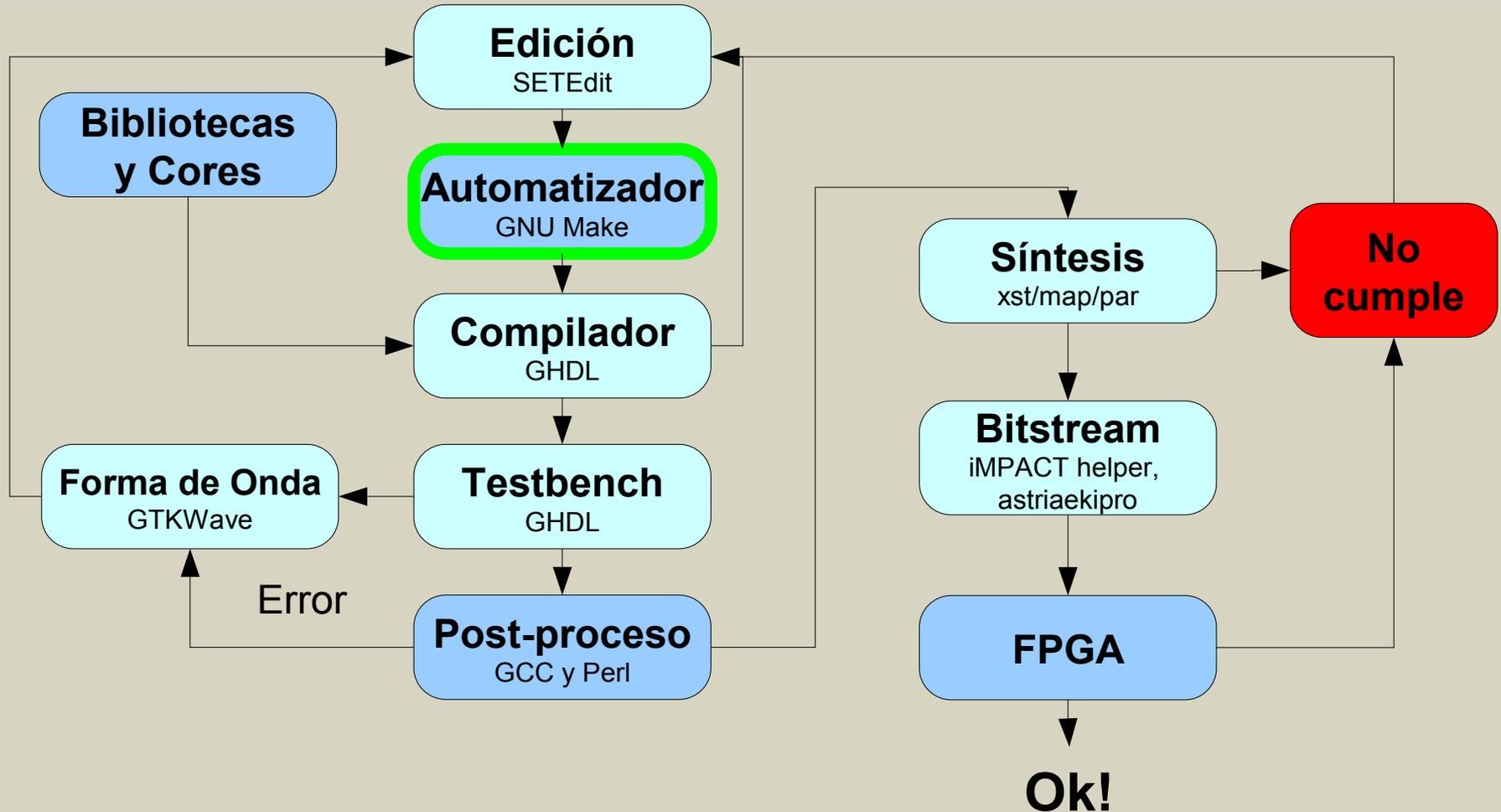
Estas son algunas de las características que lo hacen una buena elección para el trabajo con VHDL:

- Resaltado de sintaxis para VHDL.
- Macros específicas con construcciones típicas de VHDL (PMacros).
- Utilización de Exuberant C Tags con soporte específico para VHDL.
- Indentado coherente con los guidelines del proyecto.
- Configurable y con soporte para los lenguajes más populares.
- tpl2file: templates de Xilinx como archivos



<http://setedit.sourceforge.net>

## Automatización: GNU Make



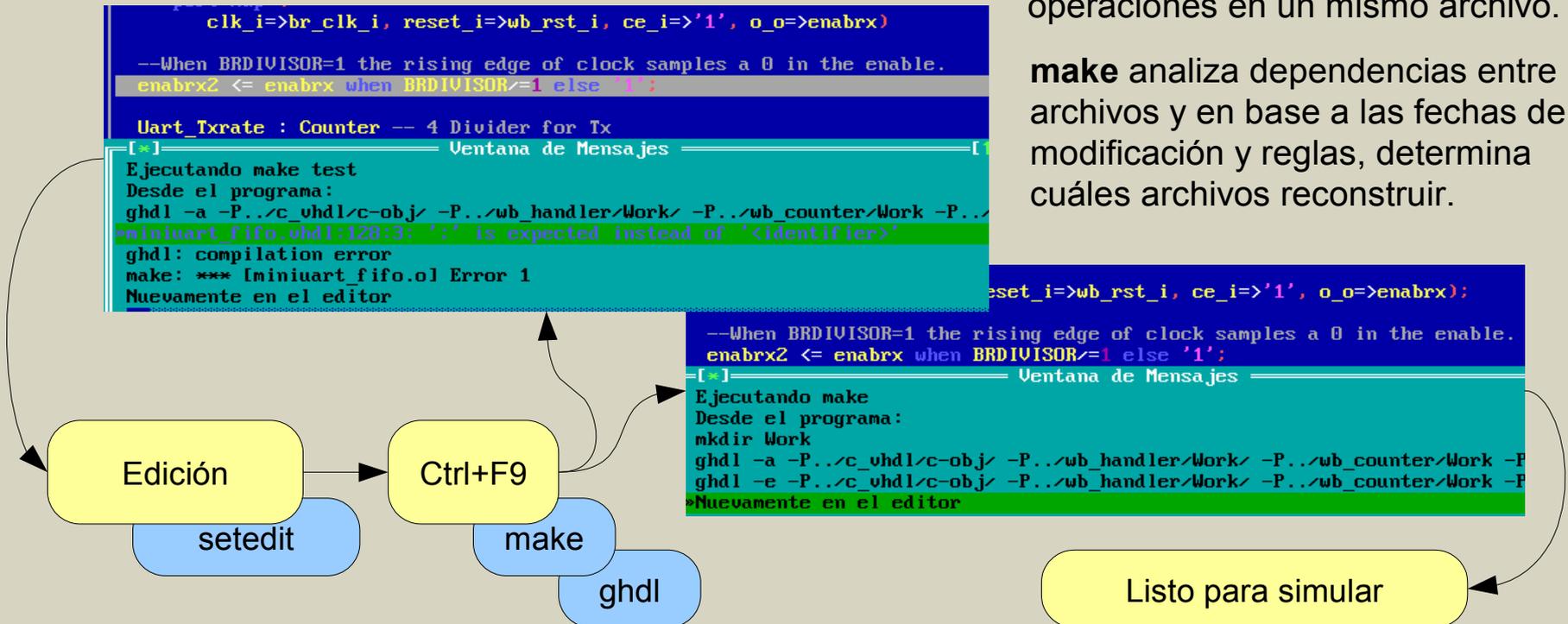
## Automatización: GNU Make

- Automatización de tareas repetitivas.
- Reducción del tiempo necesario para regenerar *targets* (objetivos).
- Integra con SETEdit para la recolección de errores.

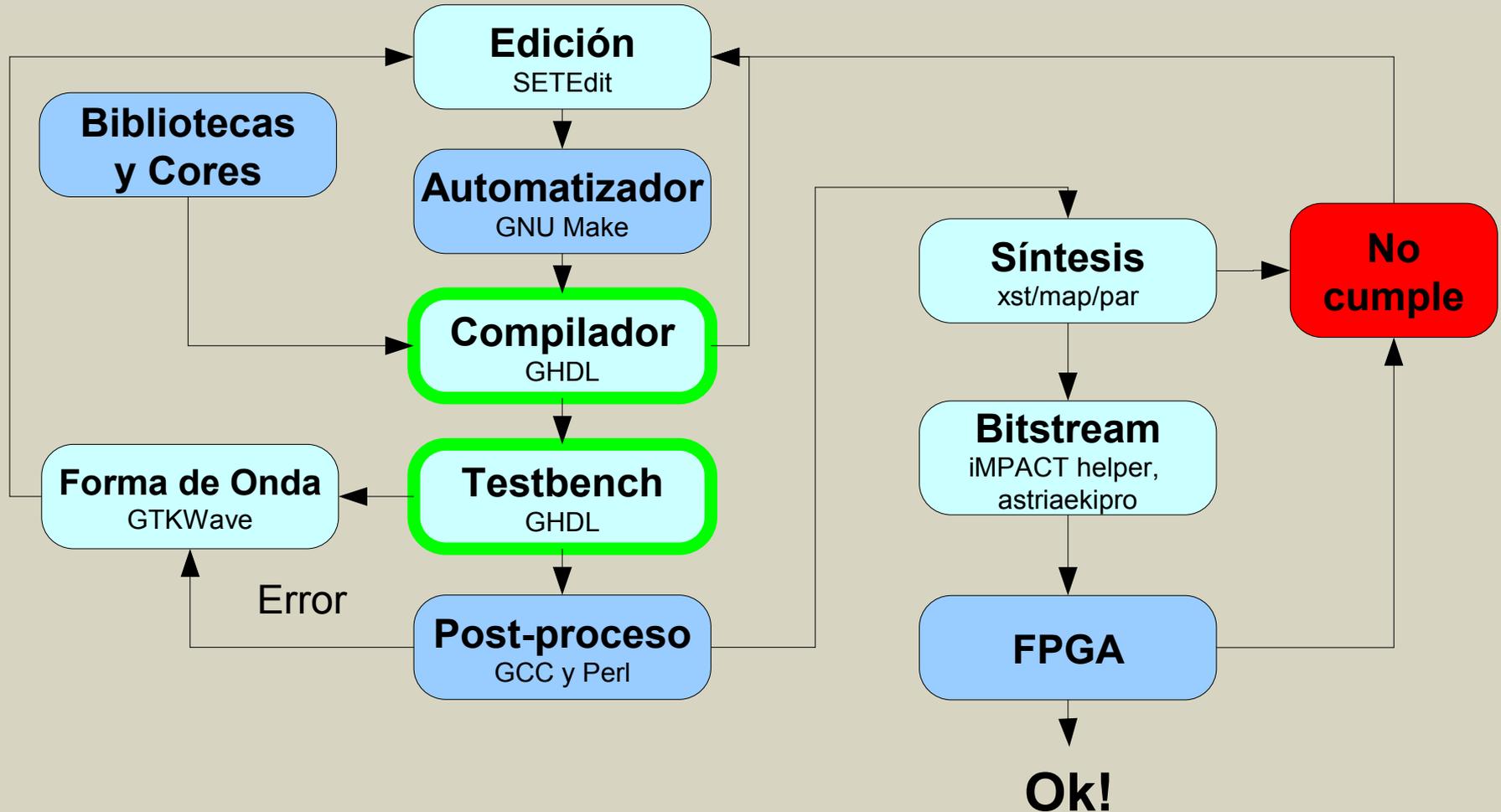
¿Por qué no un *Shell script*?

- Usualmente sólo necesitamos regenerar ciertos *targets*.
- Permite centralizar varias operaciones en un mismo archivo.

**make** analiza dependencias entre archivos y en base a las fechas de modificación y reglas, determina cuáles archivos reconstruir.



## Simulación y testbench: GHDL



## Simulación y testbench: GHDL

Simulador de VHDL, que implementa los estándares IEEE 1076-1987 (VHDL87), IEEE 1076-1993 (VHDL93) y algunas características del IEEE 1076-2000 (VHDL00}. Es Software Libre.

Compila sin problemas proyectos tales como el procesador LEON y el DLX.

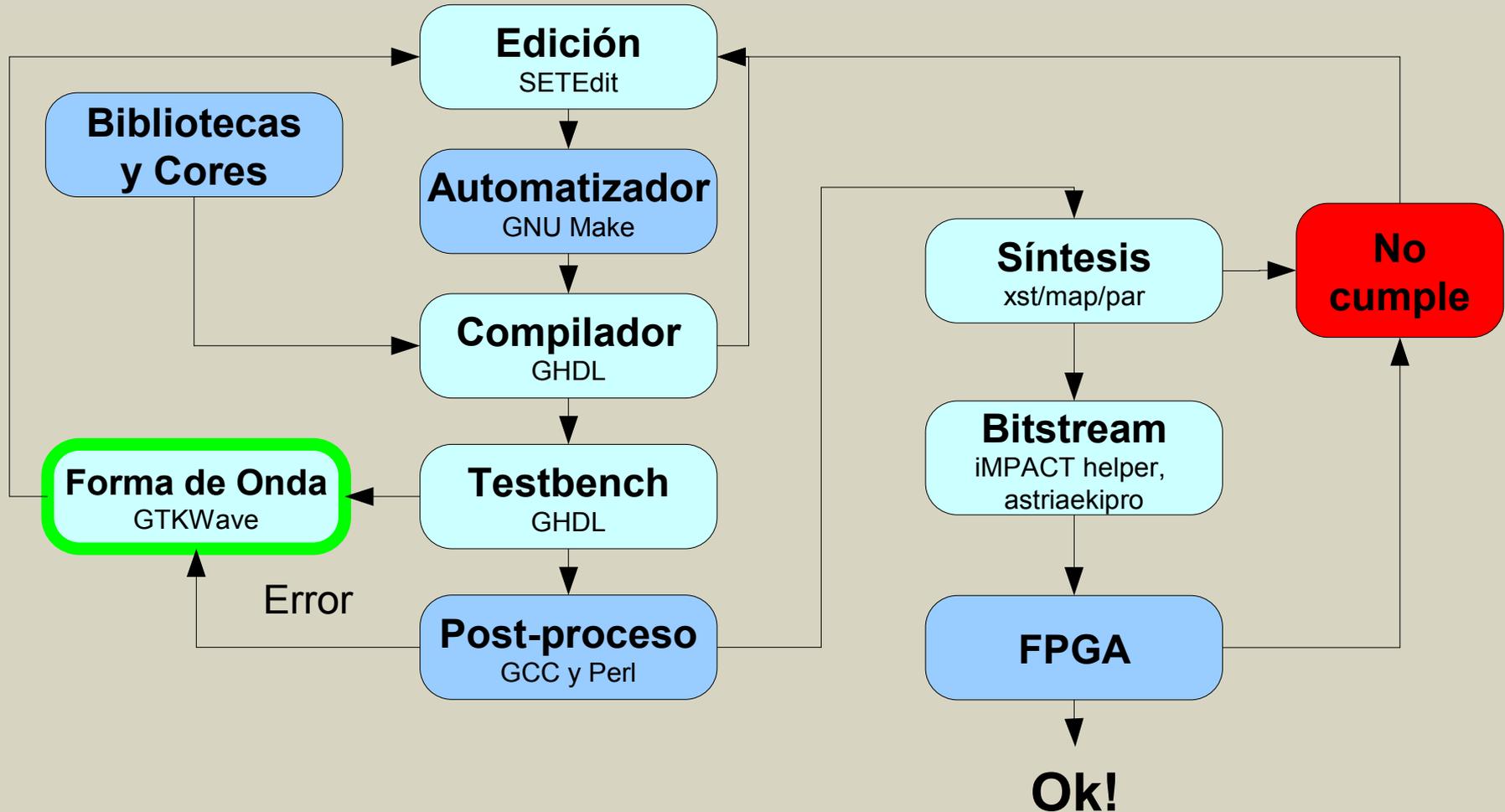
Utilizamos el GHDL como herramienta principal de simulación para VHDL.

**<http://ghdl.free.fr/>**

GHDL utiliza la tecnología del GCC, el compilador de software libre más utilizado en todo el mundo.

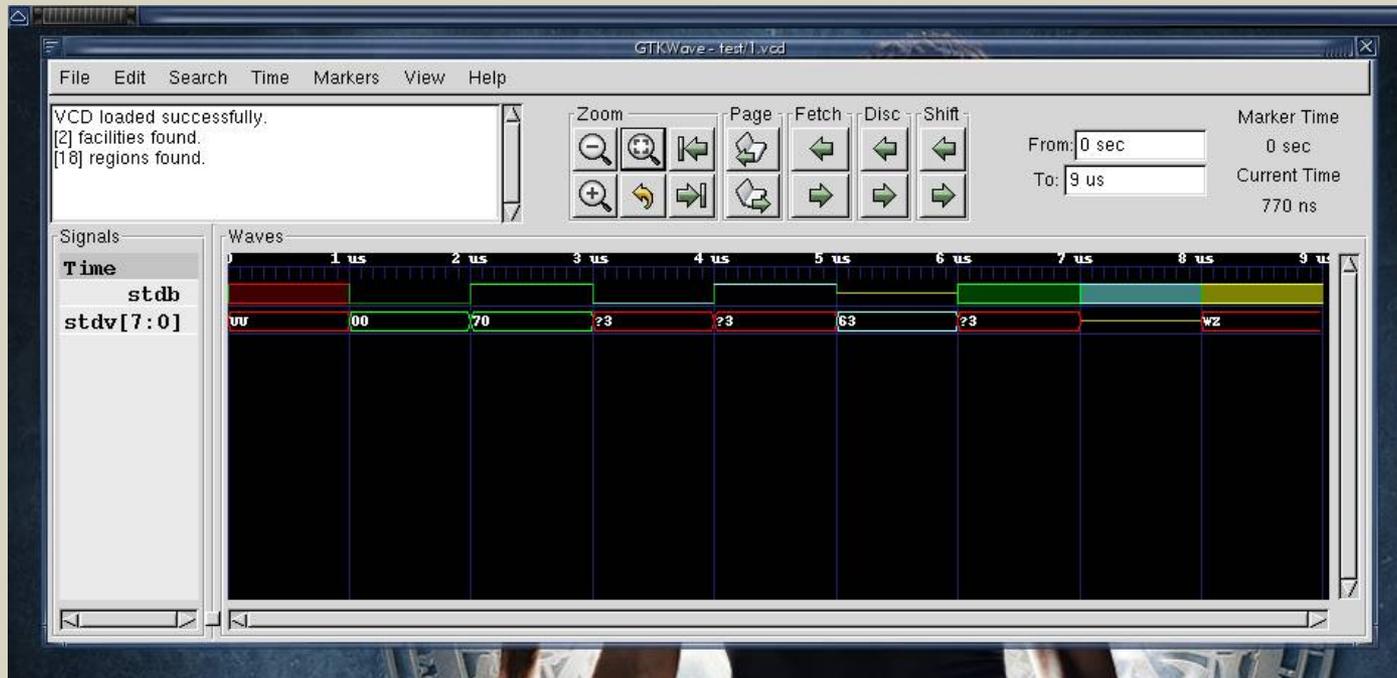


## Análisis: GtkWave

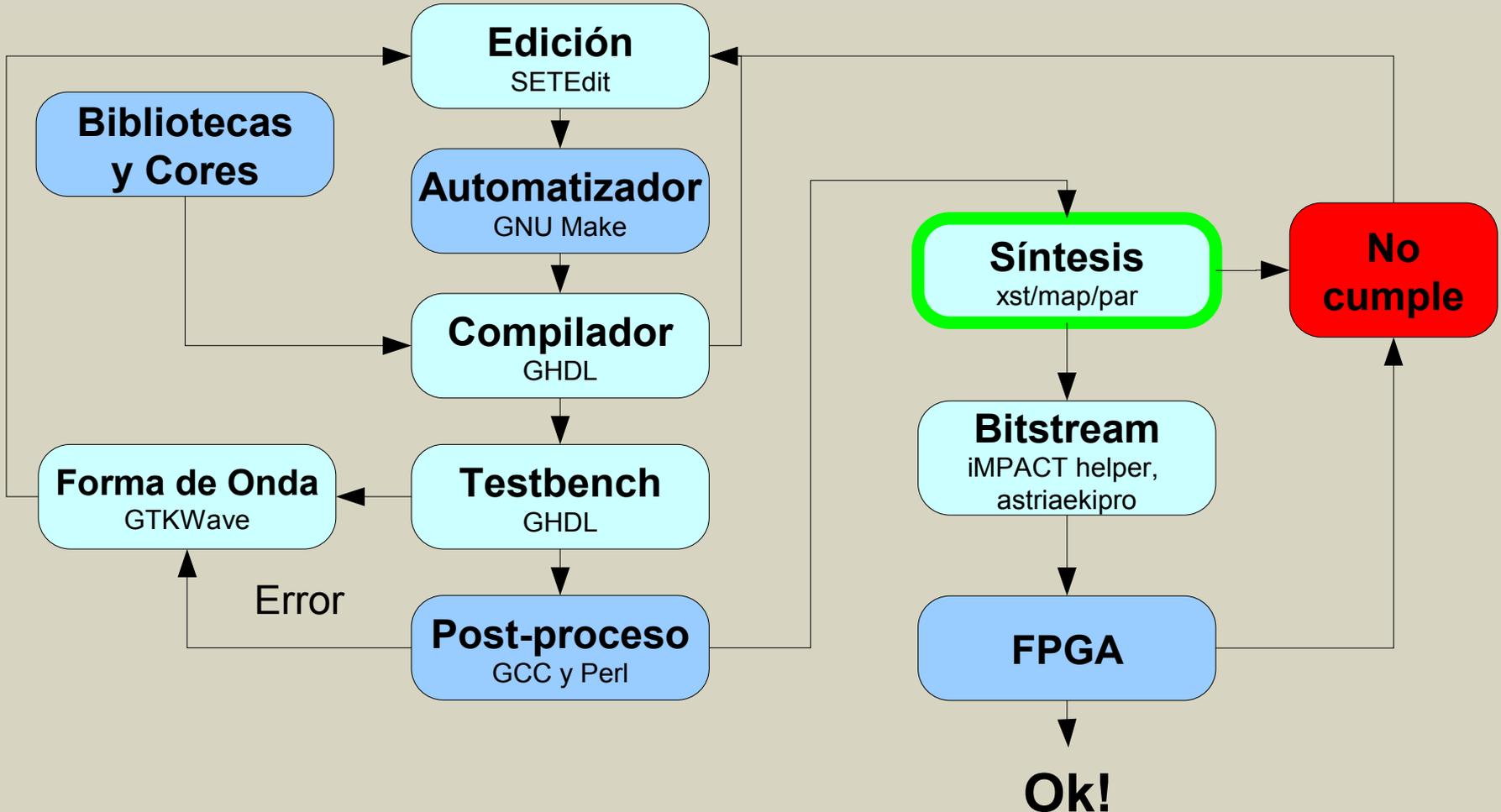


## Análisis: GtkWave

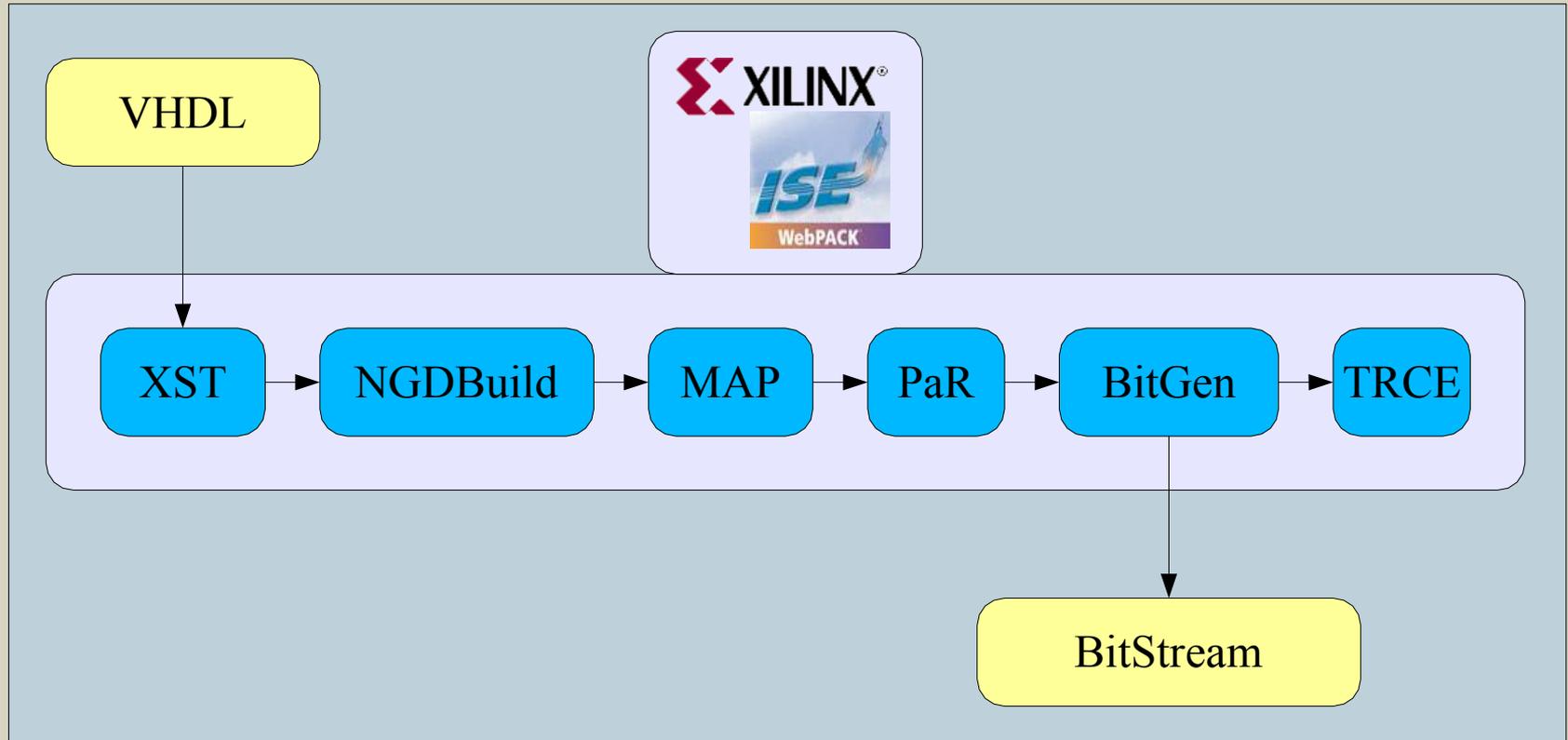
- Cuando estamos haciendo el *debugging* de un diseño es útil poder observar las formas de onda digitales. **GtkWave** es un visor de forma de ondas Software Libre.
- Interfaz de usuario simple escrita en GTK. Muy liviano.
- Soporte especial para VHDL (originalmente sólo Verilog).



## ISE WebPack



## ISE WebPack



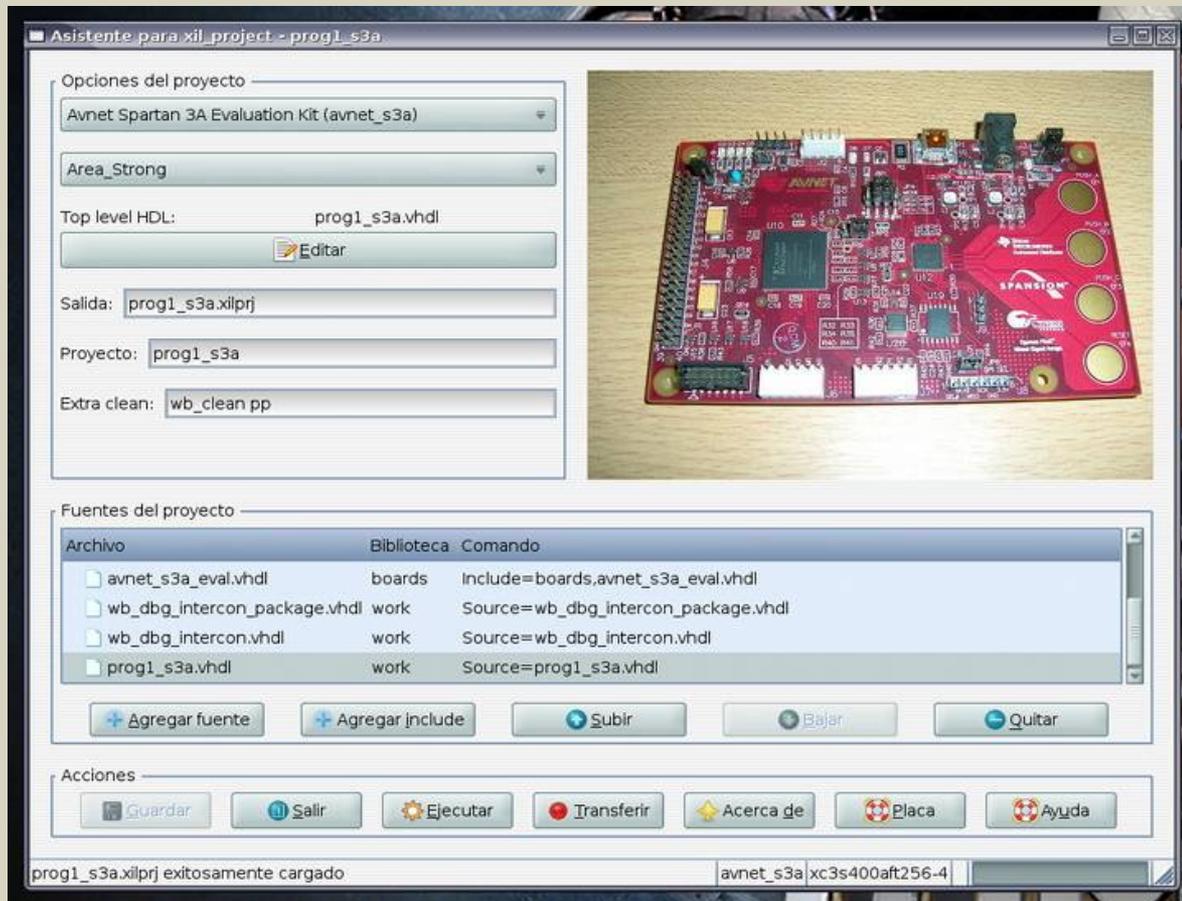
## Ayuda: Xil Project/Wizard

### xil\_project

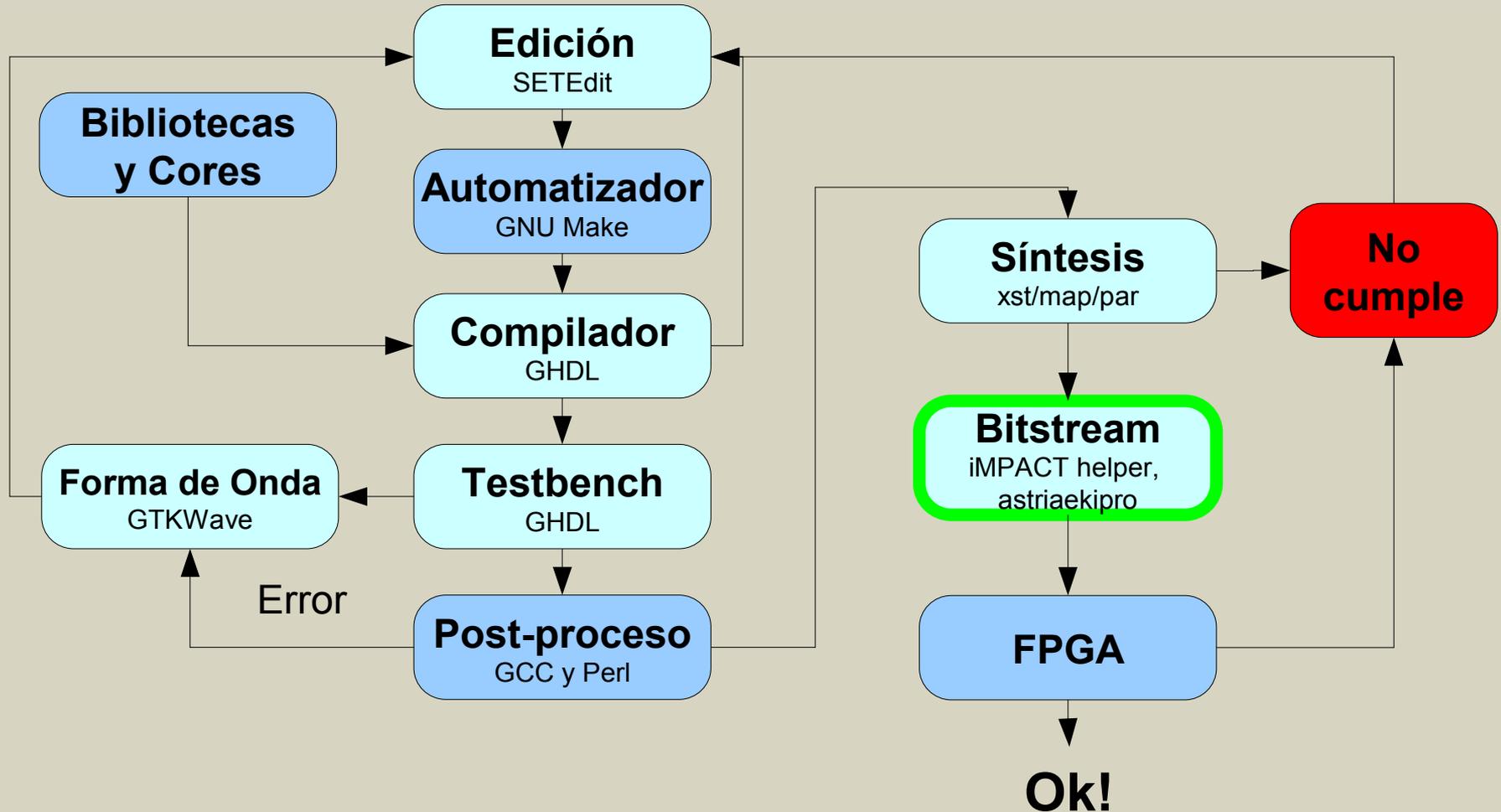
- Herramienta para facilitar la síntesis utilizando herramientas de Xilinx.
- Orientada al uso de línea de comandos y asume que se siguen los lineamientos de FPGALibre.
- Objetivo: generar archivos necesarios para la síntesis a partir de un archivo de proyecto y la descripción de la placa a utilizar.

### xil\_project\_wz

- Interfaz de usuario gráfica que ayuda con la generación del archivo de proyecto.
- Permite introducirnos fácilmente al uso de **xil\_project**.



## Transferencia del bitstream



## Transferencia del bitstream

### Comunicación más común:

- JTAG estándar IEEE 1149.1 del JTAG (Joint Test Action Group).
- Originalmente pensado para testeo.
- Independiente del fabricante.

### IMPACT Helper

- Herramientas `do_impact` y `make_impact`
- Línea de comandos (automatizable)
- Más simple.
- No necesita drivers en el kernel
- En algunos casos hay que desarrollar herramientas.

### Programador JTAG:

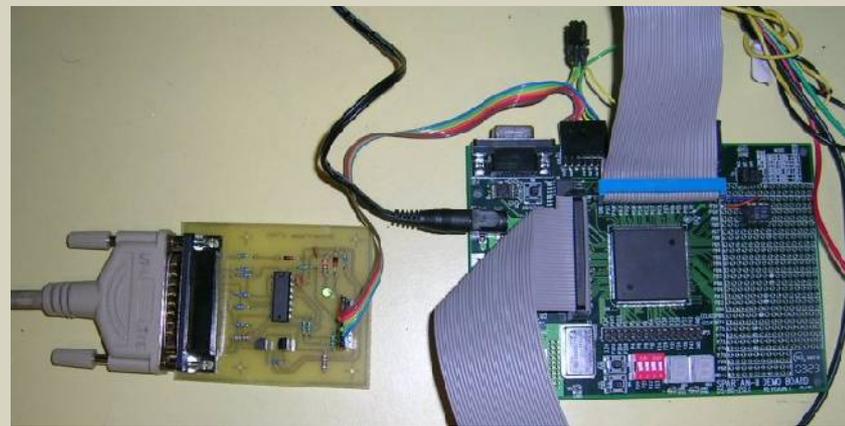
Conectan a la PC con la placa que contiene la FPGA.

Usamos el Parallel III de Xilinx, también conocido como DLC5.

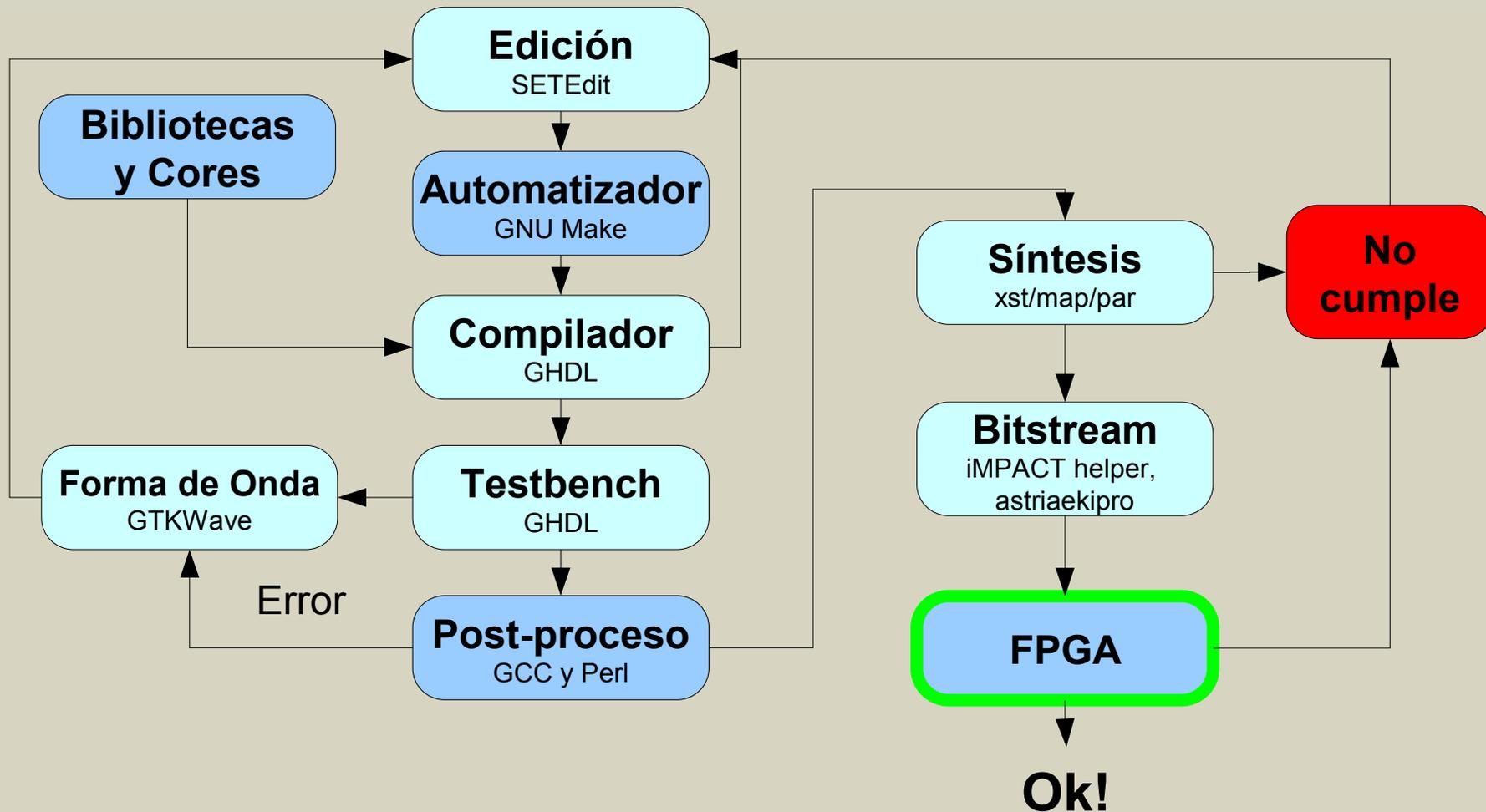
Este es un circuito simple y barato que se puede conectar al puerto paralelo de una PC.

Todos los archivos de diseño se encuentran disponibles en formato Kicad.

FPGA Libre  
Free FPGA



## Hardware : Placa de desarrollo con FPGA

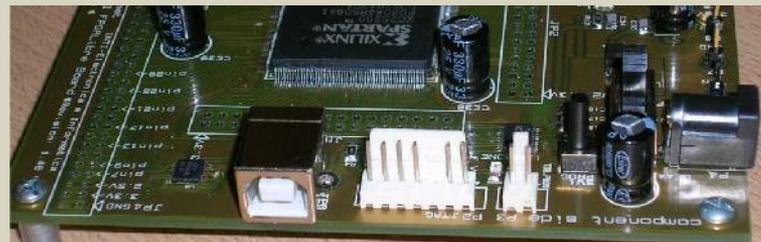


## Hardware : Placa de desarrollo con FPGA



- Para desarrollar una aplicación, además del *chip* **FPGA**, es necesaria una electrónica de soporte: circuitos impresos, circuitos de alimentación, memorias, conectores, etc.
- La forma más fácil de abordar este tema es comprando algún kit de desarrollo. También se puede encarar el desarrollo.

### S2Proto



- Parte del proyecto FPGA Libre. Diseño e implementación de un circuito impreso con FPGA, pensado para ámbitos de desarrollo e instituciones educativas. Bajo licencia GPL para permitir su libre utilización, implementación, modificación y comercialización.
- Desarrollado y probado con SL: Kicad y GNU jtag.
- Soporte para dispositivos Xilinx Spartan II PQ208.

**FPGA Libre**  
Free FPGA

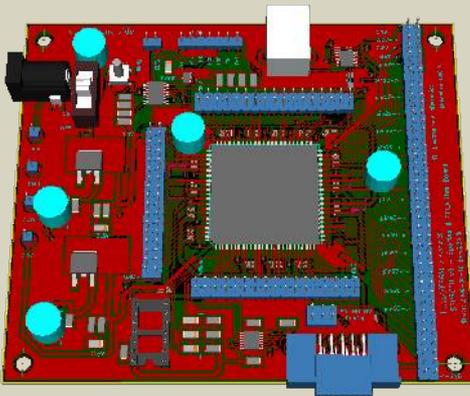
**KICAD** GPL PCB SUITE

## Diseño de circuitos impresos

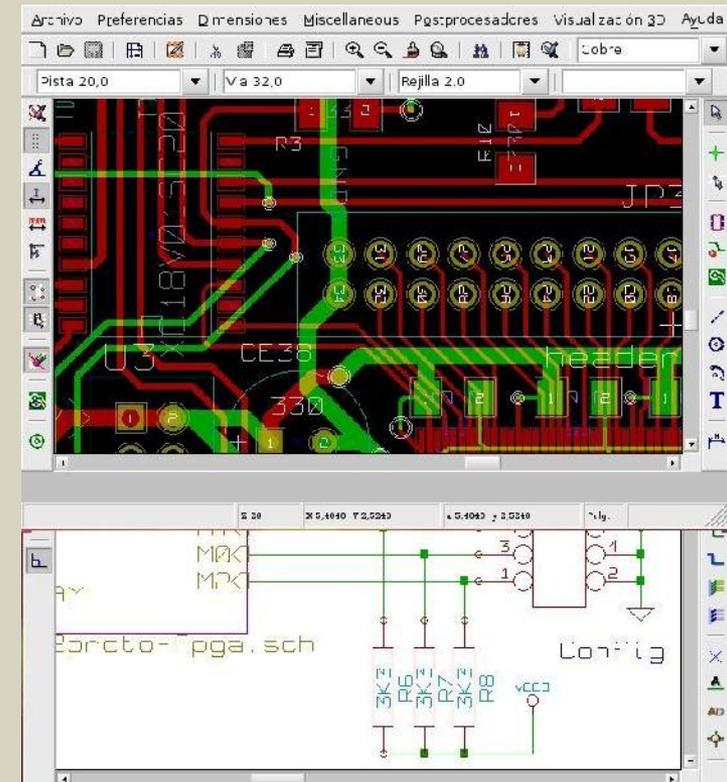
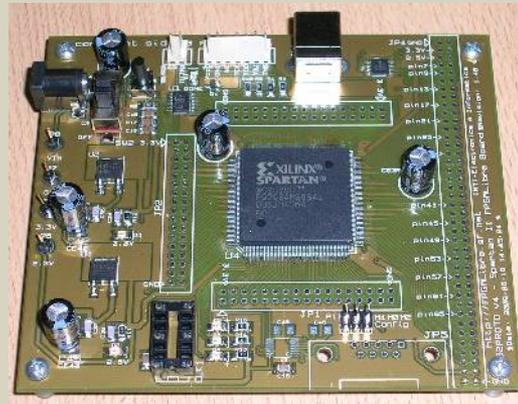
# KICAD

## GPL PCB SUITE

- KICAD
- Cubre las tres tareas principales:
  - Ingreso de circuito esquemático.
  - Ruteo de PCB (Printed circuit Board)
  - Visor de formato Gerber.



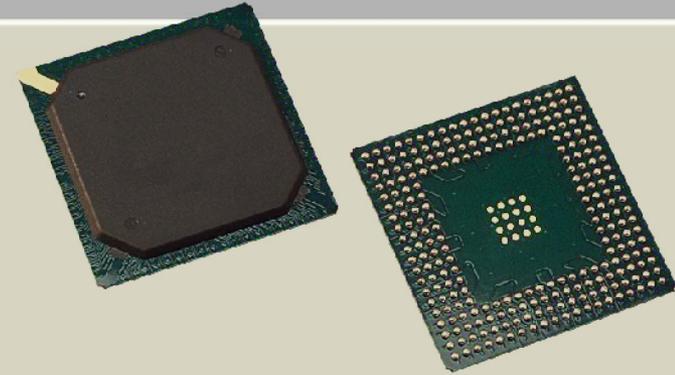
Placa S2Proto en KICAD y físicamente





## Información adicional y contacto: Proyecto FPGALibre

- Facilitar el intercambio de conocimientos y cores.
- Impulsar el uso de herramientas de S.L.
- Hosteado por SourceForge
- <http://fpgalibre.sourceforge.net>
- Abierto (OSs y tecnologías)
- Actualmente basado en Debian GNU/Linux
- Objetivos:
  - Impulsar el desarrollo con dispositivos FPGA utilizando herramientas de S.L. u Open Source.
  - Fomentar el intercambio y desarrollo de cores IP con licencias que posean el mismo espíritu que las del S.L.



**FPGA Libre**  
FPGA Libre



## Agenda

# Demostración y Consultas

- **Demostración de las herramientas. Ejemplo de como hacer titilar un led con una FPGA!!!!**
- **Consultas**



**INTI**

Instituto  
Nacional  
de Tecnología  
Industrial

**SASE 2011**  
Simposio Argentino de Sistemas Embebidos

Ministerio de Industria  
Secretaría de industria y Comercio



**¡MUCHAS GRACIAS!**

**Desarrollo con FPGAs en  
GNU/Linux**

**Jefe de Laboratorio DESoL:**  
Ing. Salvador Tropea

Av. Gral. Paz 5445 (1650) San Martín  
Buenos Aires, Argentina  
(11) 4724-6315  
{salvador,brenji,rmelo}@inti.gov.ar

<http://utic.inti.gov.ar/>  
<http://fpgalibre.sf.net/>

Marzo de 2011



Licencia de la presentación



**Atribución-SinDerivadas 2.5  
Argentina**

Usted es libre de:



**copiar, distribuir, exhibir, y ejecutar la obra**

Bajo las siguientes condiciones:



**Atribución. Usted debe atribuir la obra en la forma especificada por el autor o el licenciante.**



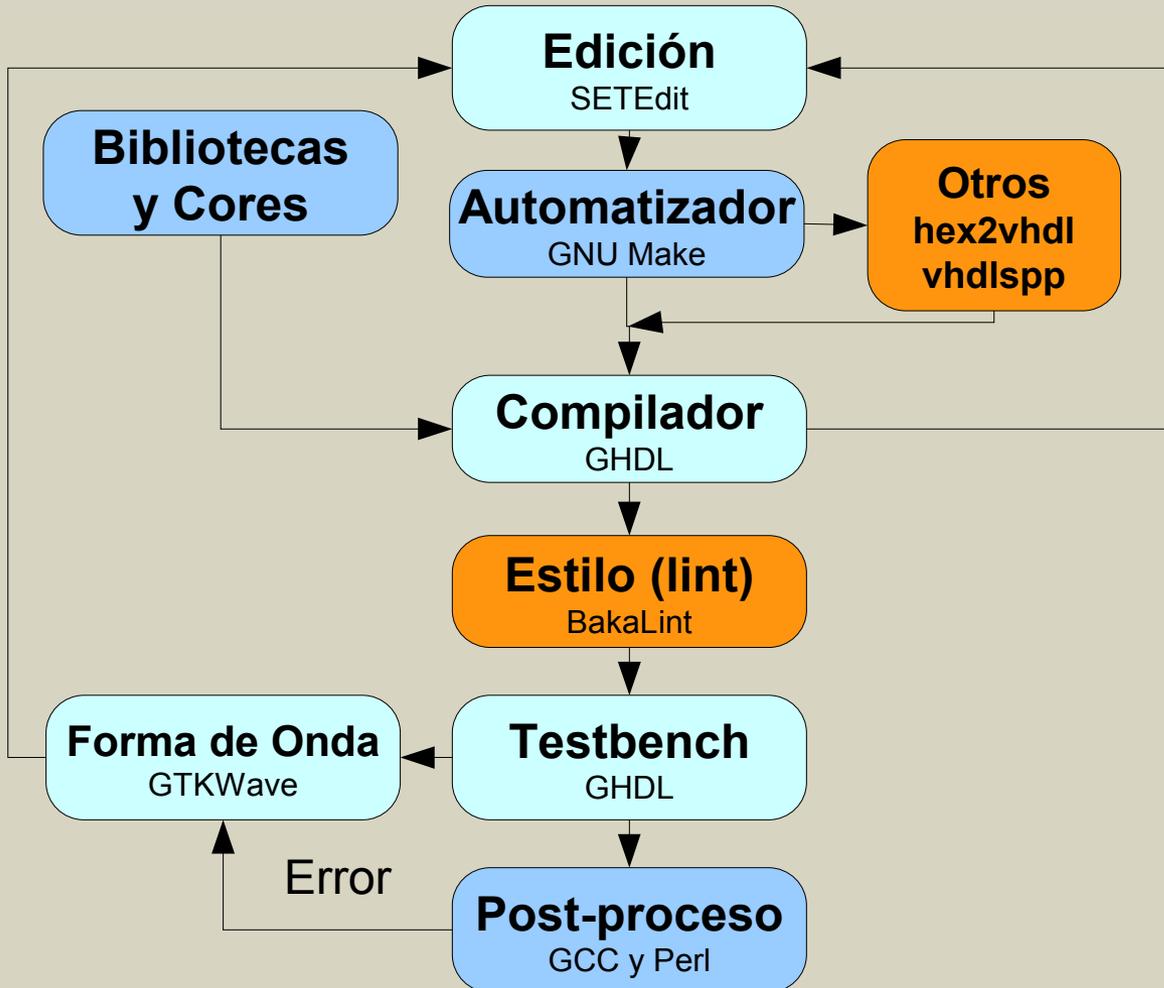
**Sin Obras Derivadas. Usted no puede alterar, transformar o crear sobre esta obra.**

<http://creativecommons.org/licenses/by-nd/2.5/ar/>

## Agenda

# Anexo

## Herramientas para FPGA - Adicionales



### hex2vhdl:

- Conversor de .hex a array VHDL (ROM).

### vhdlspp (VHDL Simple PreProcessor):

- Permite reemplazar *tags* (marcadores) del tipo `@nombre_archivo@` por el contenido del archivo indicado.

### bakalint:

- Permite verificar si el código VHDL que escribimos cumple los *guidelines* del proyecto.

FPGA Libre  
Free FPGA

# Proyectos y Trabajos

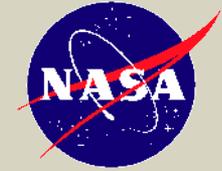
- **IP Cores desarrollados bajo GNU/Linux**
- **Kit de desarrollo S2PROTO**
- **Tecnoplac III-USB**
- **GNU/Linux embebido en procesadores LEON**

## Bloques de propiedad intelectual desarrollados y/o adaptados (2005-2010)

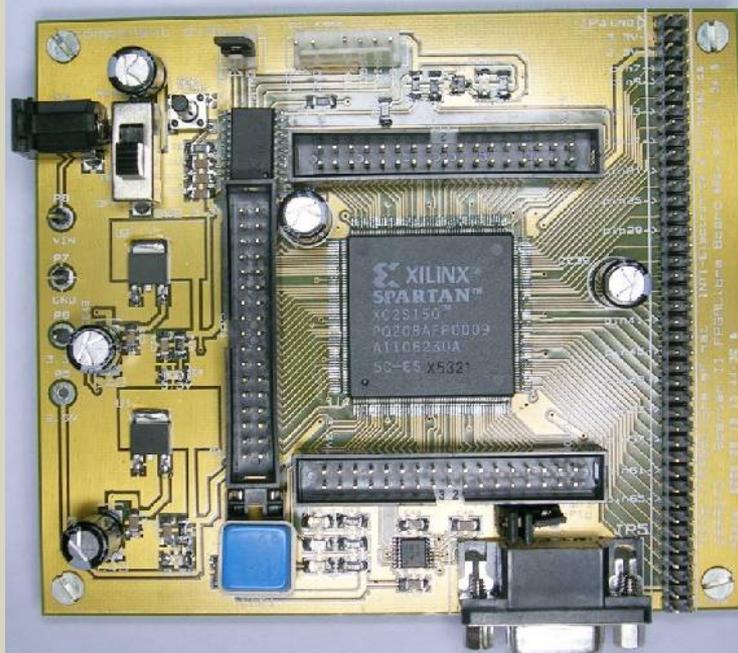
- Microprocesadores
  - PIC16C84
  - AVR (ATtiny22, ATmega103, ATmega8, ATmega32, etc.) **OC**
- Comunicaciones
  - Intrasistemas: I<sup>2</sup>C **OC**, SPI
  - Externa: PS2, USB, MAC Ethernet
- Control
  - Servomotores y motores paso a paso
  - Encoder
- Varios
  - Display alfanumérico
  - Controlador de interrupciones
- A pedido de la industria aeroespacial
  - Logaritmo decimal (43 bits)
  - Adquisición a 250 KHz 16 bits y cálculo de la varianza.
  - Contador de pulsos con ventana adaptativa.



**OC**



## Tarjeta S2proto de hardware libre para desarrollo con FPGA (2005-2006/2008)



*s2proto-v2*

### Actualización:

- Conector USB tipo B y driver (niveles estándares USB).
- Reemplazo de memoria por una más económica.

**FPGA Libre**  
FPGA Libre

**KICAD**  
GPL PCB SUITE

- Parte del proyecto FPGA Libre. Diseño e implementación de un circuito impreso con FPGA, pensado para ser utilizado en ámbitos de desarrollo e instituciones educativas.
- Brindado bajo licencia GPL para permitir su libre utilización, implementación, modificación y comercialización.
- Desarrollado y probado con SL: Kicad y GNU jtag.
- Impreso doble faz de 12x10 cm de fácil fabricación en el país y montaje manual de los componentes.
- Soporte para dispositivos Xilinx Spartan II PQ208.
- Puerto RS-232 y JTAG (configuración y testeo).
- Pines de I/O disponibles para el usuario.
- Alimentación simple de 5 V CC.

*s2proto-v4*



## TECNOPLAC3-USB - Sistema de control de calidad para leche de bebés (2004-2005/2010)

- Equipo desarrollado a pedido de la empresa Mastellone Hnos. para reemplazar un equipo importado de muy alto costo.
- Los equipos desarrollados realizan actualmente el control de calidad del 100% de la producción de leche para bebés envasada en cartones de 250 cm<sup>3</sup>.
- La PC de control utiliza sistema operativo Debian GNU/Linux y Gnome.
- La aplicación de interfaz de usuario hecha a medida utiliza Allegro, Turbo Vision y libusb entre otras.
- Las placas de la electrónica de control se realizaron en KICAD.
- Una de las placas de la electrónica de control posee una FPGA que implementa el sistema USB y la lectura de un encoder.
- Para el desarrollo en la FPGA se utilizó el ciclo de trabajo del proyecto FPGALibre.



*Equipo para control de calidad de leche para bebés en envase de 250 cm<sup>3</sup>*



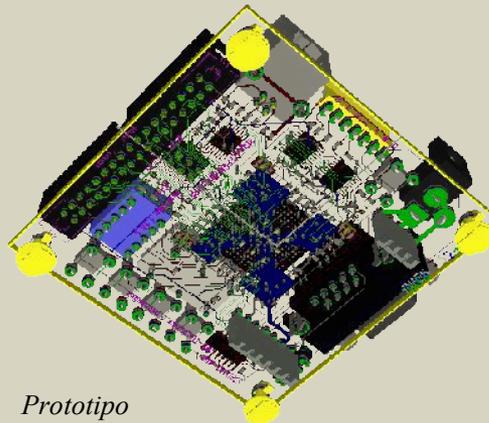
## GNU/Linux embebido en procesadores LEON (2008-2010)

- El procesador LEON es un procesador SPARC apto para FPGAs, desarrollado originalmente por la Agencia Espacial Europea. Es muy utilizado en la industria aeroespacial pero sirve para otras aplicaciones.
- La descripción de hardware del procesador y sus periféricos principales están en lenguaje VHDL bajo licencia GPL.
- Se investiga en el laboratorio esta tecnología y sus aplicaciones, tratando de cubrir las partes de software y hardware faltantes para lograr un ciclo de trabajo libre y abierto.

### FPGA Libre

#### HARDWARE:

Se está desarrollando un circuito impreso (S3Proto), que se publicará bajo licencia de hardware libre y abierto, con capacidad para correr un sistema GNU/Linux en una FPGA con LEON.



Prototipo S3Proto-Mini de 4 capas y chip BGA, realizado en KICAD y próximo a fabricarse



#### SOFTWARE:

Se desarrolló una herramienta libre: FPGALibre Leon Monitor (FLeMon) que permite interactuar con el hardware para debug y grabar una imagen Linux en la memoria Flash.



GNU/Linux y BusyBox corriendo sobre LEON3/Glib, usando una placa FPGA comercial.

