TUTORIAL

Simulación de Diseños VHDL con Software Libre: La herramienta GHDL

Juan González Gómez

Escuela Politécnica Superior
Universidad Autónoma de Madrid

IV Jornadas sobre Computación Reconfigurable y Aplicaciones, JCRA
Universidad Autónoma deBarcelona. Septiembre 2004

Índice

Introducción(I): *Software* libre

A diferencia del *software* tradicional, el *software* libre nos ofrece **4 libertades**:

- **Libertad de uso** (Cualquiera lo puede usar)
- **Libertad de modificación** (Las fuentes están disponibles)
- **Libertad de compartir** (Se puede copiar)
- **Libertad de distribuir las modificaciones** (Lo puedo mejorar y distribuir estas mejoras)

Introducción(II): ¿Por qué Software libre?

Algunas ventajas en el campo **docente e investigador**:

- **Coste cero**. No hay licencias por las que pagar
- **Mayor agilidad en la implantación/pruebas**. No hay que gestionar licencias
- **Biblioteca universal de conocimiento**
- **No reinventar la rueda**
- **No está ligado a un fabricante concreto**: potencialmente portable a cualquier plataforma
- **Ciclo de vida más largo**

Estado del arte(I)

- Campo del VHDL y simulación: **software libre muy por detrás de de las herramientas comerciales**

- No hay ninguna aplicación libre que pueda competir todavía con herramientas como **ModelSim** o **ActiveHDL**

- Sin embargo, las herramientas están los suficientemente maduras como para usarlas en entornos docentes

- La falta de "potencia" se compensa por las ventajas del Software libre

Estado del arte(II): Herramientas libres

Al más puro estilo Unix, no hay un entorno integrado que lo haga todo. Más bien, pequeñas utilidades que combinadas me permiten trabajar con el VHDL

- Editor de Texto ASCII. **Herramienta EMACS**
  - Resaltado de sintáxis
  - Uso de plantillas
- Compilador/simulador. **Herramienta GHDL**
  - Compilador basado en la herramienta GCC de GNU
- Visualizador de formas de onda: **GTKWAVE**
  - Impresión a ficheros *Postscript* para documentaciones
- Otras heramientas: **Make**

EMACS

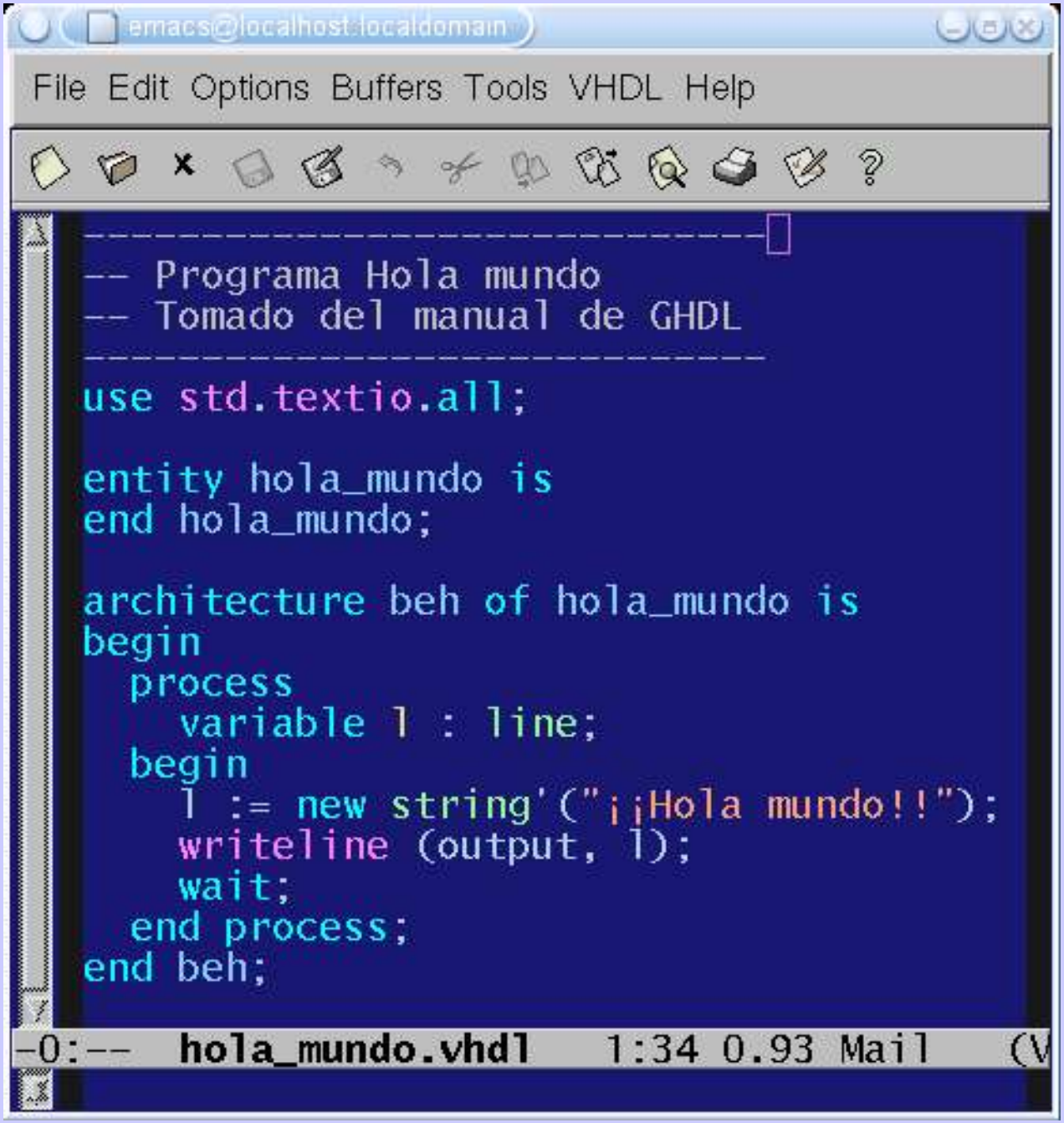
GHDL

GTKWAVE

Características GHDL (0.13)

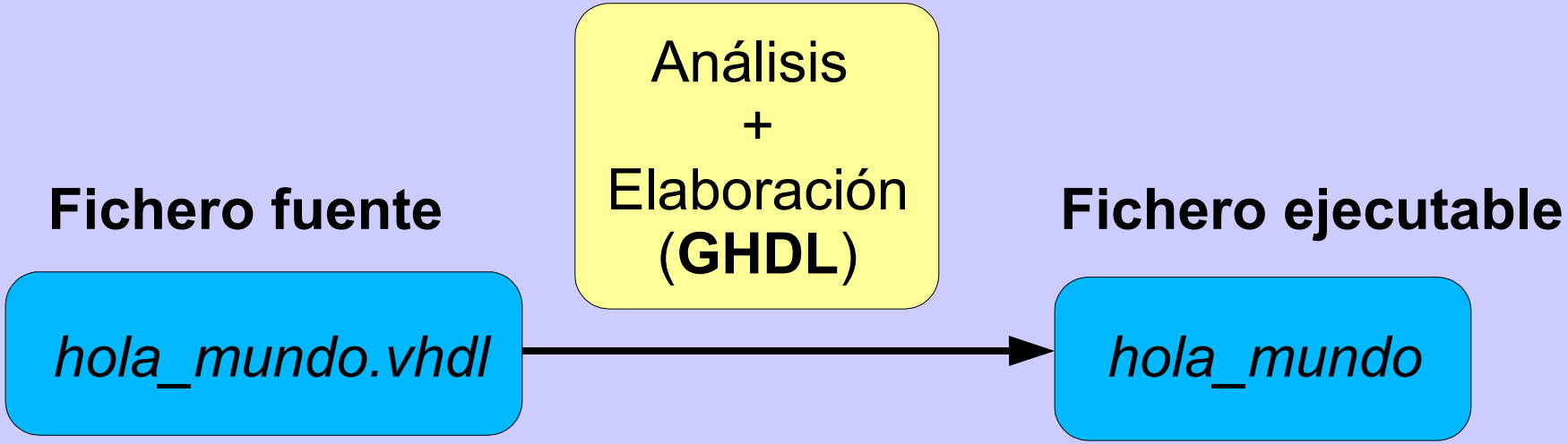
- Compilador/simulador de VHDL, basado en el **GCC** de GNU
- Es una aplicación para **consola**, no gráfica
- No utiliza lenguajes intermedios. **Traduce directamente VHDL a código máquina**.
- Genera **ficheros "ejecutables"**. Al correrlos, se realiza la simulación.
- **VHDL87** (IEEE 1076-1987), **VHDL93** (IEEE 1076-1993), **VHDL00** (IEEE 1076-2000) e incluye algunas de las revisiones realizadas en el 2002.
- Simula correctamente el procesador **DLX** y **LEON1**
- Permite la **invocación de funciones escritas en C y ADA**

Ejemplo 1: “Hola mundo”



- Se imprime la cadena "Hola mundo" por pantalla
- Editor EMACS
- Resaltado de sintáxis
- Plantillas
- Documentación
- "Asistente"

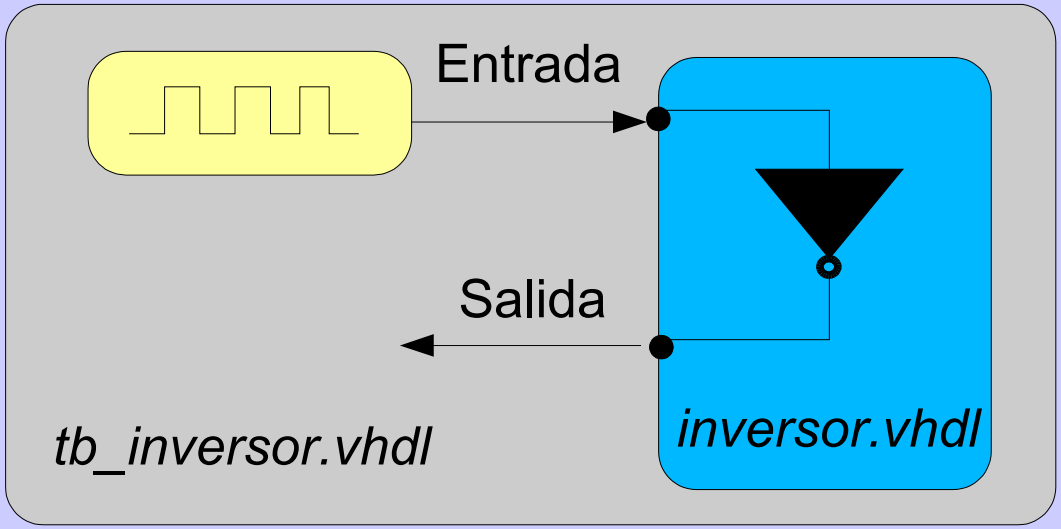
Ejemplo 1: “Hola mundo”



- **Comandos a ejecutar en la consola:**

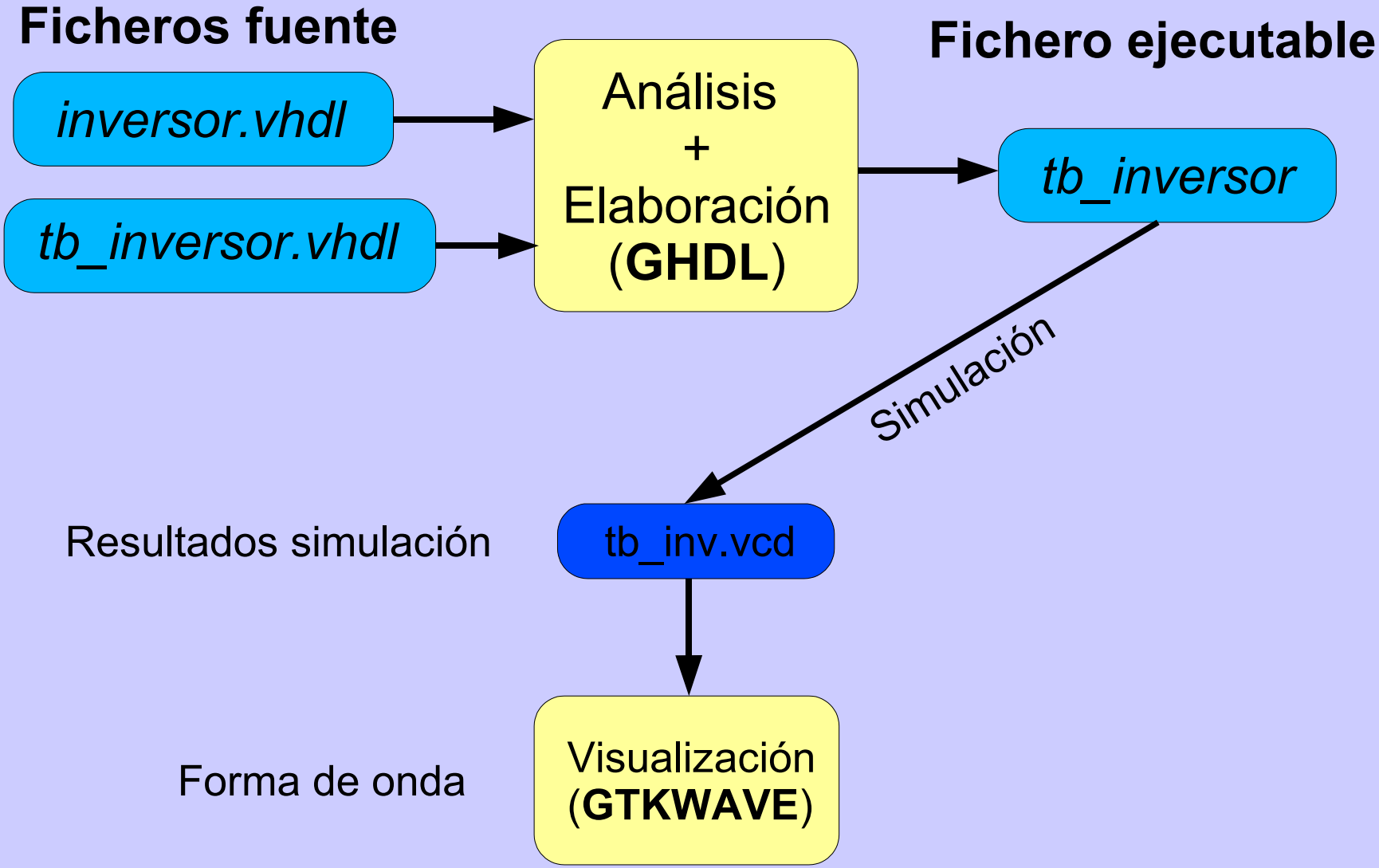
```
$ ghdl -a hola_mundo.vhdl
$ ghdl -e hola_mundo
$ ./hola_mundo
¡¡¡Hola mundo!!!
```

Ejemplo 2: un inversor



- **Entidad inversor**: fichero *inversor.vhdl*
- **Banco pruebas**: *tb_inversor.vhdl*
  - Genera una señal cuadrada por la entrada
  - Queremos visualizar la "salida"

Ejemplo 2: un inversor



Ejemplo 2: un inversor

- **Comandos a ejecutar en la consola:**

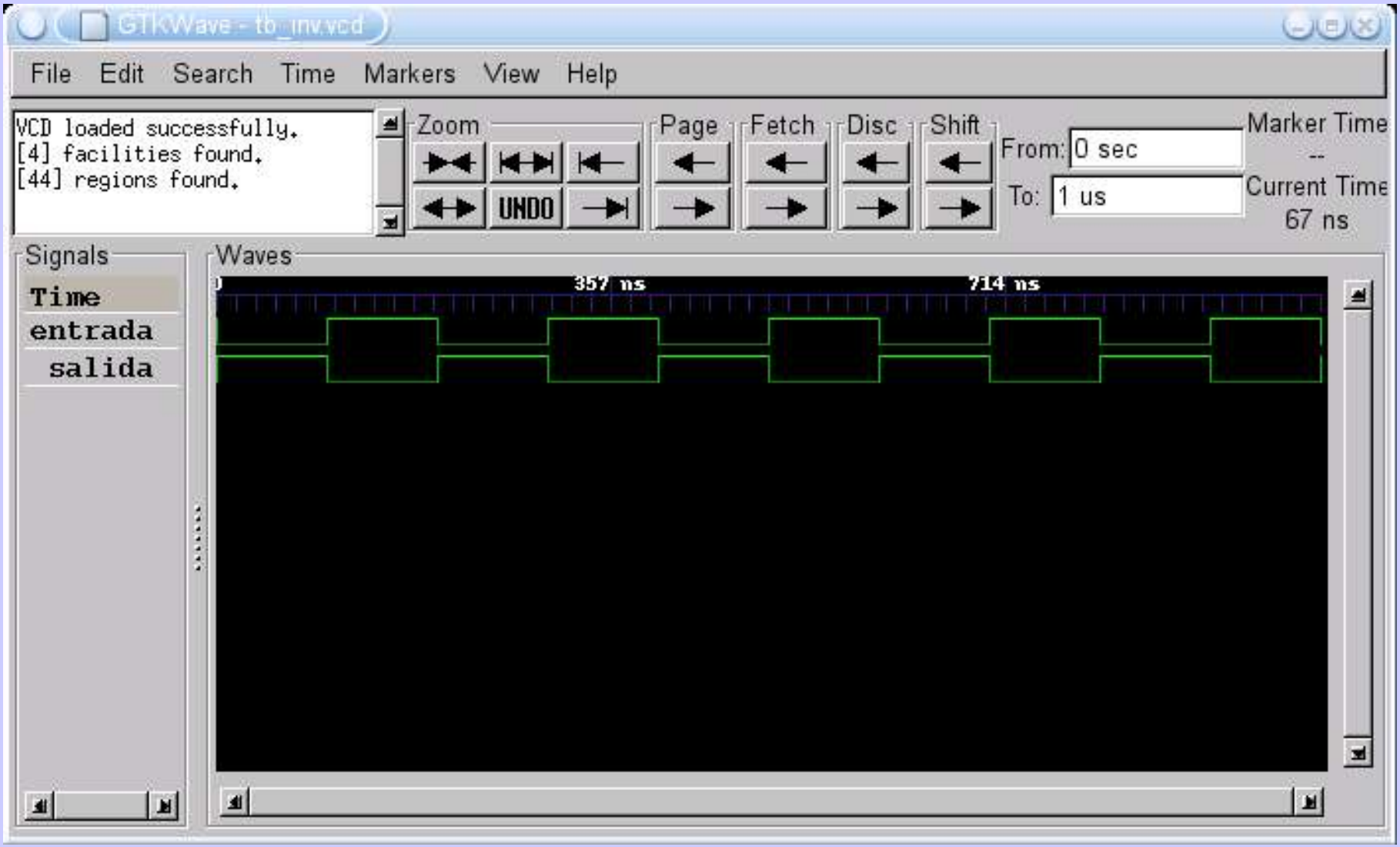
Análisis y elaboración:

```
$ ghdl -a inversor.vhdl tb_inversor.vhdl
$ ghdl -e tb_inv
```

Simulación y visualización de los resultados:

```
$ ./tb_inv --stop-time=1000ns --vcd=tb_inv.vcd
$ gtkwave tb_inv.vcd
```

Ejemplo 2: un inversor



Características avanzadas: Manejo de proyectos con make

- Si el proyecto tiene varios ficheros .vhdl, la "compilación a mano" es muy pesada.
- GHDL puede crear ficheros *Makefile*, que contienen las relaciones de dependencia entre las entidades
- Los diseños se pueden simular utilizando la herramienta *make*

```
$ ghdl -i *.vhdl
$ ghdl --gen-makefile tb_inv > Makefile
```

Importar entidades a librería de trabajo

Generar *Makefile*

Para obtener el ejecutable se invoca *make*:

```
$ make
```

Sólo se analizarán los ficheros .vhdl modificados

Características avanzadas: Invocación de funciones C, desde VHDL

- **Para hacer Bancos de pruebas nos puede interesar acceder a funciones en C:**
  - Ej. Generar datos aleatorios
  - Ej. Funciones matemáticas: sin, cos, pow...

Ejemplo "hola mundo"

Paquete en vhdl

Función en C

test.vhdl

```
package test is
 function holac return integer;
 attribute foreign of holac :
 function is "VHPIDIRECT holac";
end test;

package body test is
  function holac return integer is
  begin
   assert false severity failure;
  end holac;
end test;
```

holac.c

```
#include <stdio.h>
int holac(void)
{
 printf ("Hola desde C...\n");
 return 1;
}
```

Características avanzadas: Invocación de funciones C, desde VHDL

hola_mundo.vhdl

```
use std.textio.all;
use work.test.holac;

entity hola_mundo is
end hola_mundo;

architecture beh of hola_mundo is
begin
  process
    variable l : line;
    variable v : integer;
  begin
    l := new string'("Hola desde VHDL...");
    writeline (output, l);
    v:=holac;
    wait;
  end process;
end beh;
```

Se invoca a la función en C

Características avanzadas: Invocación de funciones C, desde VHDL

La "compilación manual" sería:

```
$ ghdl -a test.vhdl
```
Análisis del paquete

```
$ ghdl -a hola_mundo.vhdl
```
Análisis del programa principal

```
$ gcc -c -o holac.o holac.c
```
Compilación del fichero C

```
$ ghdl -e -Wl,holac.o hola_mundo
```
Elaboración y enlazado con función en C

Ejecución del programa:

```
$ ./hola_mundo
Hola desde VHDL...
Hola desde C...
```

Conclusiones

- Las herramientas libres para simulación de VHDL no tienen la "potencia" de las herramientas comerciales, sin embargo, **están lo suficientemente maduras como para su uso docente**

- **Tienen las ventajas derivadas del software libre**: longevidad en el tiempo, multiplataforma, ninguna gestión de licencias...

- El GHDL incluye **algunas características interesantes**, no soportadas por las herramientas comerciales:

  - Acceso a funciones escritas en C y ADA

  - Generación de un ejecutable

Enlaces de interés:

- **GHDL**: http://ghdl.free.fr
- **GTKWAVE**: http://www.cs.man.ac.uk/apt/tools/gtkwave/index.html
- **LEON1**: http://www.estec.esa.nl/wsmwww/leon/leon.html
- **DLX**: http://www.csee.umbc.edu/courses/undergraduate/411/spring96/dlx.html

TUTORIAL

Simulación de Diseños VHDL con Software Libre: La herramienta GHDL

Juan González Gómez

Escuela Politécnica Superior
Universidad Autónoma de Madrid

IV Jornadas sobre Computación Reconfigurable y Aplicaciones, JCRA
Universidad Autónoma deBarcelona. Septiembre 2004