

Simuladores para los proyectos de VHDL

J. Resano, J.L. Briz, EINA - Univ. Zaragoza

Febrero 2023

Los proyectos se pueden realizar con cualquier simulador de VHDL. En este documento hemos incluido información de cómo instalar algunos de ellos. Esta información es cambiante porque ha habido cambios de normativas en los Estados Unidos de América que afectan a las licencias gratuitas de ModelSim.

1 GHDL & GTKWave

GHDL es una cadena de herramientas de código abierto para analizar /compilar /montar /simular diseños en VHDL. Está disponible para Linux, MacOS y MS Windows. Se maneja desde el intérprete de comandos (en el caso de MS Windows Cygwin, CMD o PowerShell). Puedes bajar los fuentes, gestionarlo con `git` y compilarlo, o bien instalarlo como paquete en Debian /Ubuntu o MacOS. En MS Windows no hemos localizado un paquete fiable directamente instalable por lo que te recomendamos compilarlo, o bien aún mejor usar una distribución Linux en máquina virtual (e.g. Oracle VM VirtualBox). Para Mac users GHDL+GTKWave es la única opción que conocemos para simular VHDL. Para el resto, es la opción más simple y ligera.

GHDL empaqueta por defecto un generador de código para MacOS/x86 (`mcode`). Para Linux'es hay que usar `llvm` o `gcc`. En las instrucciones que damos aquí asumimos Linux/x86 e instalamos la variante `ghdl-llvm`.

GHDL no es un entorno integrado (aunque sí integrable) y para visualizar las formas de onda es preciso utilizar un visualizador independiente. GTKWave, como GHDL, se instala fácilmente como paquete en cualquier Linux, MacOS y Windows.

1.1 Instalación (Ubuntu Linux)

```
1 sudo apt-get update
2 sudo apt-get install ghdl-llvm
3 sudo apt-get install gtkwave
4 ghdl-llvm --disp-config
```

Este último paso simplemente comprueba la instalación. Observa que en `/usr/bin` se han creado los ejecutables `ghdl` y `ghdl-llvm`¹.

1.2 Workflow básico

Hay muchas formas de proceder con `ghdl` pero te aconsejamos este *workflow* básico. Partimos del directorio en el que tienes los fuentes `vhdl`. Asumimos extensión `vhd` tipo `windows`, que la ENTITY con el testbench se llama `testbench`, y que en ese directorio has creado un subdirectorío para el proyecto al que llamamos `WORK`, para no embarullar el directorio en el que tienes tus fuentes de partida.

```
1 ghdl -i --ieee=synopsys -fexplicit --workdir=WORK *.vhd
2 ghdl --gen-makefile --ieee=synopsys --workdir=WORK testbench > Makefile
```

En el paso 1 estás creando el proyecto en el directorio (`WORK`). Las opciones `--ieee` y `-fexplicit` hacen falta solo si estamos usando la librería `float` de `ieee`; cosas que pasan cuando dejas la guardería². En el paso 2 creamos un fichero `Makefile` que nos permitirá hacer modificaciones a los ficheros y volver a construir un ejecutable simplemente ejecutando `make`. Por razones desconocidas `gen-makefile` no sabe pasar la opción `-fexplicit` al `Makefile`, por lo que (en caso de que necesites usar la librería `ieee`) tendrás que editar dicho `Makefile` y añadir esa opción a la variable `GHDLFLAGS` para que luzca tal que así:

```
GHDLFLAGS= --ieee=synopsys -fexplicit --workdir=WORK
```

Ahora ya puedes ejecutar

```
1 make
2 ./testbench --stop-time=500ns --wave=test.ghw
3 gtkwave test.ghw &
4 ghdl --clean --workdir=WORK
```

En el paso 2 ejecutamos el ejecutable; ajusta el tiempo al que necesites para simular completamente el testbench o programa de prueba que tengas en cada caso. El último paso limpia todo, si se van embarullando los cambios. En caso de pánico `rm -r WORK`, y vuelves a crear el proyecto.

Añadir nuevos fuentes al proyecto es fácil con `-i` pero tendrás que regenerar el `Makefile`. Eliminarlos significa modificar el `Makefile` a mano o bien borrar el fuente y regenerar el `Makefile`. Por eso lo mejor es que guardes las RAMs fuera del directorio en el que compilas, que mantengas en este último solo las dos RAMs (datos e instrucciones) siempre con el mismo nombre (e.g. `RAM-D.vhd`, `RAM-I.vhd`), y que para cambiar de RAMs (e.g. cuando tienes que cambiar de programa de prueba) hagas algo parecido a:

```
\$ cp ../rams/ram-datos-que-quiero RAM-D.vhd
\$ cp ../rams/ram-instr-que-quiero RAM-I.vhd
```

¹`ghdl` es un driver al segundo; un alias `ghdl="ghdl-llvm"` haría el mismo efecto

²O antes: cuando los papás te abandonan en ella.

1.3 Lugares de ayuda

1. [GHDL en GitHub](#).
2. [GTKWave en GitHub](#)
3. [Landing page de GTKWave aconsejable para Mac users](#).
4. [Nuestro tutorial rápido en Moodle \(YouTube\)](#)
5. [El mejor manual que encontrarás de GHDL](#).
6. `man ghdl` (en Linux)

2 ModelSim Microchip

- **Registrarse** accediendo a [este enlace](#). Usad el correo de unizar.
- **Descargar** el entorno. Para ello seleccionar la pestaña Software en el enlace anterior (Fig. 1), localizar la versión de la Fig.2 y descargar.
- Descomprimir el .zip; no hay que instalar nada. El ejecutable está en `modelsim\modeltech\win32acoem`.
- **Obtener la licencia** en [este enlace](#). Clickar en *Register a free license* (Fig. 3) y descargar la licencia (Fig. 4). Si os dan distintos tipos de licencia para elegir, para un equipo de Windows la adecuada es: *Liberio Evaluation Node Locked License for Windows*. Para solicitarla os pedirán el número de serie de vuestro disco duro C:. En la propia ventana os explican como obtenerlo haciendo `Vol C:` en una ventana de comandos de Windows. Todo esto es para generar una licencia que sólo funcionara en el equipo que tenga ese disco duro. La licencia os llegará al correo en unos minutos.
- **Copiar la licencia** (`license.dat`) en el directorio `C:\flexlm`.
- Comprobar que la variable `LM_LICENSE_FILE` está definida en las variables de entorno y añadir la dirección de la licencia: `c:\flexlm\License.dat`.

3 ModelSim - IntelFPGA

Intel también permite descargar ModelSim desde [este enlace](#).

- Seleccionar *Intel Quartus Prime Pro Edition* (e.g. la 21.2)
- Seleccionar descargar *Individual Files* y a continuación *ModelSim-Intel*FPGA*,
ficheros:

Figure 1: Página para la descarga de ModelSim

ModelSim ME v10.5c Stand-Alone Software Release (08/03/2017)

ModelSim ME v10.5c for Windows®

Figure 2: Página para la descarga de ModelSim

- ModelSim - Intel FPGA Edition (includes Starter Edition)
- ModelSim - Intel FPGA Edition (includes Starter Edition) Part 2
- Instalar el fichero como administrador. Para la versión 21.2.0.72 en Windows ejecutar el fichero `ModelSimProSetup-21.2.0.72-windows.exe`
- Aparece una ventana en la que hay que seleccionar *Modelsim Starter Edition*, que no precisará archivo de licencia.
- Aceptar los términos de licencia y seleccionar directorio de instalación

4 Active-HDL Student Edition

La empresa Aldec tiene un simulador para Windows muy potente que sirve, entre otros, para VHDL. El entorno es parecido a modelsim. Se puede descargar en [este enlace](#). Os pedirán que os registréis (hay que hacerlo con el correo de unizar) y os enviarán un enlace para descargarlo.

5 Xilinx ISE

El entorno ISE Webpack de Xilinx está disponible para Windows y Linux. Su descarga requiere registrarse en Xilinx. En Windows 10 se ejecuta sobre una máquina virtual Linux y en total ocupa unos 38 GB:

- Es conveniente instalar primero Oracle MV VirtualBox.
- Si existe, el instalador de Xilinx lo reconoce, y pregunta si se desea utilizar esa versión y deja tres enlaces en el escritorio de Windows 10, de los que sólo necesitarás el *Project Navigator*.

License Selection

Libero Paid and DirectCore Licenses – Can be purchased through our online purchasing portal

Paid licenses are available as NodeLocked and Floating based. Renewal Licenses are discounted per seat. Incremental licenses can be added to floating licenses during the year for pro-rated pricing.

We offer multiple licenses to design with our FPGA and SoC design tools. Most of the software tools and FPGA IP cores are freely available but a few high-value IP cores and resources needed to work with high-density FPGAs required paid licenses.



Figure 3: Registro de licencia

Seats	Registration Date	Caducidad	Download License
1	19-Jan-2022	20-Mar-2022	Download License

Figure 4: Descarga de licencia

- Si ya tienes en tu Windows un virtualizador como Oracle VM Virtual Box lo reconocerá y dará la opción de utilizarlo. Si no lo tienes, instala también el software de virtualización.
- Al clicar sobre *Project Navigator* se inicia la máquina virtual, que bota un Linux con su escritorio incluido, y a continuación bota también automáticamente el ISE.

Hemos detectado algunos pocos casos sobre Windows 10 Home Edition en los que el procedimiento anterior ha dado problemas. En ese caso, puedes crear primero una MV con VirtualBox, a partir de cualquier distribución Linux (Ubuntu o la que quieras), y descargar e instalar luego sobre esa MV la versión de ISE para Linux

No obstante, instalarlo en Windows como máquina separada tiene dos ventajas:

1. Puedes iniciar toda la cadena desde el enlace al escritorio de Windows.
2. La creación de una carpeta compartida entre el la MV Linux virtualizado y el Windows anfitrión es imprescindible. Este paso está integrado en el proceso de instalación de ISE para Windows 10.

5.1 Instalación del ISE Webpack

1. Registrarse en XILINX, mejor con el correo de la universidad, entrando en:
 - <https://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html>
2. Elegir la opción *Download ISE WebPACK software for Windows and Linux*.
3. Elegir la versión 14.7
4. Elegir el instalador (hay para Windows y Linux)
5. Descargarlo y descomprimirlo.
6. En la carpeta descomprimida encontraréis el **xsetup** que instala el ISE:
 - (a) Os pedirán que aceptéis todas las licencias
 - (b) Luego hay que elegir la versión del ISE. Elegid ISE WebPACK. El resto son todas de pago.
 - (c) En *opciones de instalación* podéis desactivar las tres opciones (*Acquire a License Key, Install WinPCap, Install Cable Drivers*)
 - (d) Elegid el directorio de instalación. Pide unos 17GB de espacio. En mi ordenador he visto que ocupa unos 11 o 12. La instalación tardará unos minutos.

6 Otros entornos alternativos

Xilinx tiene el entorno Vivado, orientado al trabajo con las FPGAs de la marca. A diferencia del ISE, Vivado recibe actualizaciones y no necesita máquina virtual en el caso de Windows. El inconveniente es que es un entorno más complejo y voluminoso, con muchas otras herramientas además del simulador de VHDL (incluso más que el ISE). La ventaja es que permite implementar tu diseño para una FPGA moderna, y ver el resultado en área, tiempo de ciclo...

Finalmente, para hacer simulaciones pequeñas se puede usar la herramienta online [edaplayground](https://www.edaplayground.com/). La ventaja de esta herramienta es que no hay que instalar nada. Pero tiene varias desventajas: no sirve para proyectos grandes (como los de AOC2) y los mensajes de error no siempre son claros.