

técnicos, la encapsulación de los recursos en una única máquina virtual y que prácticamente todos los programas son software libre.

Por otro lado, al tener instalada una distribución basada en linux, se tiene la ventaja de el aprendizaje sencillo de la plataforma libre, que puede ser necesaria en un momento dado en algún desarrollo de sistemas embebidos con un microkernel linux en un microcontrolador o un kernel linux en un procesador de aplicación. De este modo, la máquina virtual puede servir como aprendizaje al entorno linux que es tan necesario conocer hoy en día en el área de los sistemas integrados.

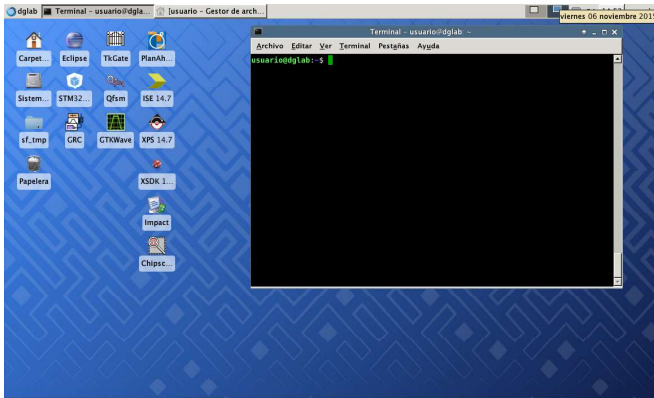


Figura 1. Laboratorio virtual dglab.

IV. APLICACIONES DISPONIBLES

El laboratorio virtual se compone no sólo de la máquina virtual, sino también de un conjunto de aplicaciones preinstaladas y una serie de tutoriales y ejemplos de uso que sirven de ayuda al alumno en el proceso de aprendizaje del uso de las herramientas contenidas en el laboratorio. En cuanto a las aplicaciones instaladas se integran dos ámbitos principales, el ámbito de los *sistemas digitales* y el ámbito de la *programación de microcontroladores* destinados a sistemas embebidos.

Todas las aplicaciones cubren los recursos necesarios dentro de las asignaturas de Electrónica Digital, Diseño Digital y Sistemas Integrados del Grado en Ingeniería en Electrónica y Automática (Rama Industrial) de la Escuela de Ingenierías industriales de la Universidad de Extremadura. Además se incluyen recursos adicionales, para que el alumno pueda familiarizarse con otras tecnologías y plataformas de desarrollo. Como objetivo, se pretende proporcionar una herramienta atractiva que despierte la curiosidad del alumno por el desarrollo de sistemas digitales y de sistemas embebidos.

Adicionalmente, estas herramientas pueden ser utilizadas por alumnos que realicen el Trabajo de Fin de Grado en los campos anteriormente mencionados. Esto es una gran ventaja para el alumno, puesto que la mayoría de herramientas de programación y desarrollo se han integrado en un único entorno, y por tanto no es necesario lidiar con nuevas herramientas ni

la instalación de las mismas, que en algunos casos puede no ser trivial.

Como entorno de desarrollo se ha escogido Eclipse por ser ampliamente utilizado y conocido dentro de la comunidad de desarrolladores de software libre para programación en Java y otros lenguajes como C o C++.

IV-A. Aplicaciones enfocadas al diseño de sistemas digitales

En el primer área de aplicación de dglab se encuentran aplicaciones destinadas al diseño de sistemas digitales mediante lenguajes de descripción hardware y simulaciones de los sistemas descritos. Para ello, en la asignatura de Electrónica Digital se proporciona un primer contacto al alumno con este mundo utilizando la aplicación TkGate. Esta aplicación utiliza el lenguaje Verilog para las descripciones del hardware objetivo de diseño. En cursos posteriores, se utiliza Eclipse como un entorno de desarrollo más profesional, junto con herramientas adicionales con el mismo objetivo, pero con mayor potencia y portabilidad de diseños de mayor envergadura.

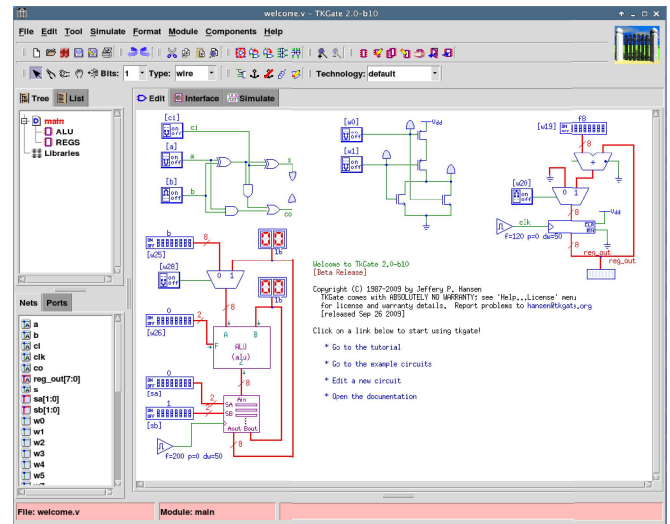


Figura 2. TkGate, Simulador de sistemas digitales.

IV-A1. TkGate: TkGate es un editor y simulador de circuitos digitales cuya interfaz se basa en Tcl/Tk. Incluye una serie de dispositivos digitales integrados como memorias, puertas lógicas, y módulos disponibles para ser editados en forma de esquemático o mediante Verilog como lenguaje de descripción de hardware.

El diseño de circuitos digitales se divide en módulos compuestos de puertas lógicas y elementos digitales simples para desarrollar la función deseada. La edición de tales módulos se puede llevar a cabo tanto en modo gráfico como con Verilog.

Al igual que en la mayoría de programas dedicados al diseño digital, a partir de un módulo definido en Verilog o mediante edición de esquemáticos pueden crearse múltiples instancias en el mismo documento.

La principal ventaja de este programa radica en el apoyo en esquemáticos del diseño. Los alumnos pueden generar módulos descritos en Verilog que se muestran gráficamente

en un editor de esquemáticos para la interconexión de dichos módulos, existiendo algunas herramientas gráficas que facilitan la simulación de los circuitos sin necesidad de introducir al alumno el concepto de un testbench o banco de pruebas.

Es por tanto, una herramienta muy útil para iniciarse en el diseño de sistemas digitales, si bien la sintaxis de los módulos Verilog no cumple por completo el estándar Verilog *IEEE 1364*.

Otra limitación, es el paso de parámetros de unos módulos a otros, estando disponibles únicamente en los módulos descritos en Verilog (mediante una sintaxis no estándar según el *IEEE 1364*), y limitando las posibilidades de un diseño parametrizable más completo.

IV-A2. Eclipse: Eclipse es un entorno de desarrollo integrado (IDE) desarrollado por la comunidad de código libre. Es un entorno muy común utilizado tanto por desarrolladores de software en proyectos de C/C++ y Java para la arquitectura x86 como en sistemas embebidos destinados a la programación de microcontroladores y microprocesadores de más bajo consumo y propósito más específico, como aquellos que comparten la arquitectura ARMv7, o algunos más simples como los microcontroladores PIC o ATMEL de 8 bits.

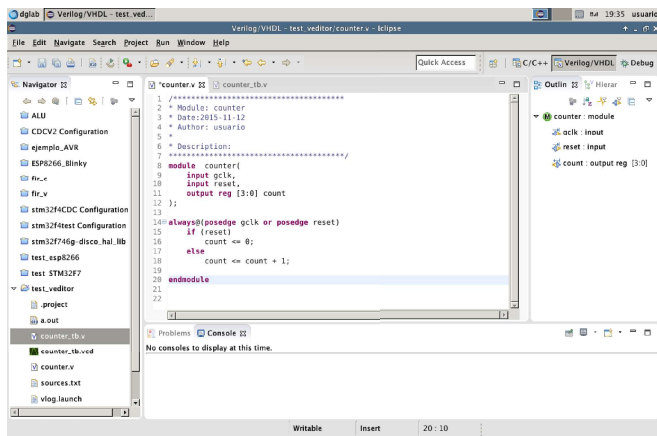


Figura 3. Eclipse y el plugin Veditor para Verilog

En el entorno de desarrollo de sistemas digitales, existen plugins para editar código en lenguajes de descripción de hardware Verilog y VHDL. En dglab, se ha incluido un plugin de software libre denominado *VEditor*[16]. Este plugin se encarga de resaltar errores de sintaxis que puedan existir en los ficheros de código. Además, la vista Verilog/VHDL permite observar las señales contenidas en cada fichero de descripción de hardware, los módulos instanciados y demás en una estructura tipo árbol muy intuitiva.

Eclipse es por tanto una alternativa a TkGate en cuanto a la edición de código Verilog, pero por sí solo no podría simular el hardware.

Para proveer a Eclipse de capacidad de simulación se utiliza el simulador *Icarus Verilog*, también una herramienta de software libre que se encarga de simular las fuentes verilog mediante un módulo *testbench* necesariamente.

IV-A3. Icarus Verilog: Icarus Verilog[17] es un simulador de fuentes Verilog compatible con el estándar *IEEE 1364*. Opera como un compilador, traduciendo la descripción de hardware (fichero *.v* y submódulos) en un formato objetivo, entre los que se incluye un formato *.vvp* que puede ser ejecutado posteriormente para realizar la simulación.

La simulación por sí misma no produce ningún resultado si no se utiliza en el diseño del testbench el comando *\$dumpvars* para producir un volcado de las variables deseadas a un fichero con extensión *.vcd*. Opcionalmente se puede optar por producir una salida por consola que informe sobre el estado de las señales en forma de texto. Este último modo de simulación permite la automatización de la simulación, evitando en muchos casos la necesidad revisión de las señales por parte del diseñador, puesto que esta funcionalidad se puede integrar directamente en el testbench. Se presenta de este modo a los alumnos el concepto de test de unidades automatizado (Automated Unit Test).

Un pequeño script permitirá ejecutar Icarus Verilog desde Eclipse.

```
find -name "*.v" > sources.txt
iverilog -f sources.txt
vvp a.out
```

El script utiliza Icarus Verilog para simular el proyecto y posteriormente ejecuta la simulación mediante *vvp*, que generará, en caso de estar configurado *\$dumpvars*, el fichero *.vcd*.

IV-A4. GTKWave: Para la visualización de los resultados, se integra un visor de formas de ondas en dglab. Este visor toma el fichero *.vcd* y lo interpreta para representar sus resultados en el visor de formas de onda.

En general, todas estas herramientas (integradas en Eclipse) proporcionan una visión más realista al alumno sobre el diseño de sistemas digitales a la que proporciona TkGate. Por tanto, están pensadas para ser utilizadas en el curso de Diseño Digital. De este modo, el alumno comprende fácilmente el flujo de trabajo iniciándose en TkGate y, posteriormente, realizar el cambio a Eclipse resulta una transición natural y que permite mayor flexibilidad.

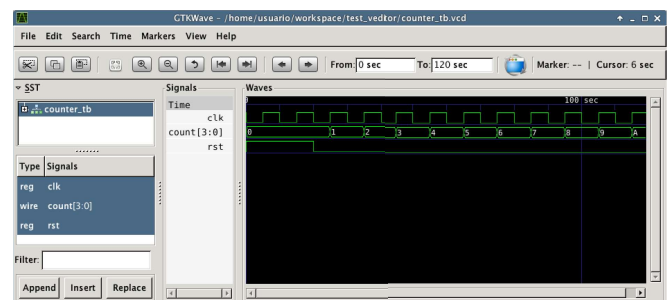


Figura 4. GTKWave, visor de formas de onda.

IV-A5. Qfsm: Para facilitar al alumno el diseño de máquinas de estado, e introducirle en los conceptos relativos a los sistemas secuenciales, se utiliza el programa *Qfsm*. Permite

la edición gráfica y la simulación de las máquinas de estado. Además, es capaz de generar ficheros VHDL y Verilog con una descripción hardware de la máquina de estados producida, lista para ser integrada en un diseño más complejo.

Con esta herramienta, se ayuda al alumno a comprender el concepto de máquinas finitas de estado, observando detalles sobre las diferencias entre autómatas de Mealy y autómatas de Moore. Por otro lado, el rápido flujo de trabajo de Qfsm permite al alumno comprobar el buen funcionamiento de una máquina de estado muy fácilmente.

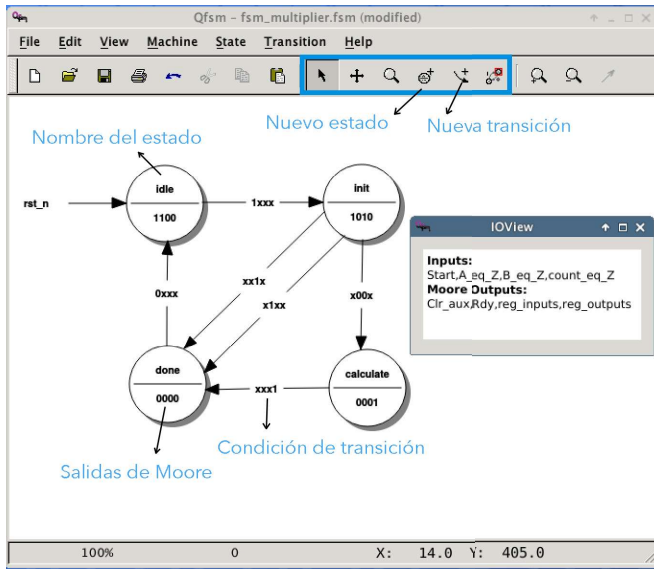


Figura 5. Qfsm, editor de máquinas de estado.

Otra ventaja de la introducción al alumno de las máquinas de estado mediante Qfsm es que el código VHDL y Verilog generado servirá como pauta y modelo de diseño de cómo debe codificarse una máquina de estados. Qfsm también proporciona al alumno la posibilidad de comprobar la integridad de la máquina de estados, en cuanto a condiciones de transiciones no ambiguas, existencia de un estado final, códigos de estado no ambiguos, etc. En el caso de encontrar uno de estos errores, Qfsm lo detecta y avisa al usuario de un error en la codificación de la máquina de estados.

IV-A6. GHDL: Aunque el principal soporte al aprendizaje se proporciona mediante el lenguaje de descripción de hardware Verilog, GHDL[18] es un programa de terminal de comandos que permite al alumno simular fuentes en VHDL.

El uso de Verilog como lenguaje de descripción de hardware supone un aprendizaje más rápido de la sintaxis, al ser más similar a un lenguaje de programación del estilo de C. Cuando el alumno ha obtenido una buena comprensión sobre el funcionamiento en paralelo del hardware y maneja correctamente las descripciones de hardware en Verilog, se le puede introducir al lenguaje complementario de VHDL. A pesar de ser bastante más verboso, lo comprenderá con muchos menos problemas, al haber comenzado con un lenguaje más similar a los que se encuentra acostumbrado.

Gracias a que el aprendizaje que el alumno desarrolla no está ligado a la sintaxis, la transferencia de estos conocimientos a otro lenguaje no supone un problema. El objetivo del docente, por tanto, no es enseñar a utilizar una plataforma de diseño, sino proporcionar un entendimiento de los sistemas digitales en sí.

Para facilitar la transición entre ambos lenguajes de descripción de hardware, se anima al alumno a utilizar *Icarus Verilog* para realizar traducciones automáticas desde Verilog a VHDL. De este modo, el alumno puede observar con detalle las diferencias en la sintaxis y observar en qué detalles deberá tener especial cuidado.

IV-B. Aplicaciones enfocadas a la programación de microcontroladores

En el segundo área de aplicación de dglab se encuentran aplicaciones destinadas a la programación de sistemas de microcontroladores, microprocesadores y en general diseño de sistemas integrados.

Las herramientas integradas en esta sección tienen como objetivo docente facilitar el diseño para diferentes plataformas, compartiendo el mismo entorno de desarrollo de Eclipse. Esto facilita al alumno el aprendizaje de la plataforma de microcontrolador, ya que, al conocer Eclipse, no necesita invertir recursos temporales en manejar el nuevo entorno de desarrollo, y le permite resolver problemas sobre la plataforma en cuestión más rápidamente.

Las diferentes plataformas integradas en dglab tienen un fin estratégico. Para captar la atención y el interés del alumno dglab permite programar microcontroladores atmel, que son el corazón de la mayoría de Arduinos. Se presentan herramientas más potentes y con fines más específicos, microcontroladores de mayores capacidades y con propósitos más específicos.

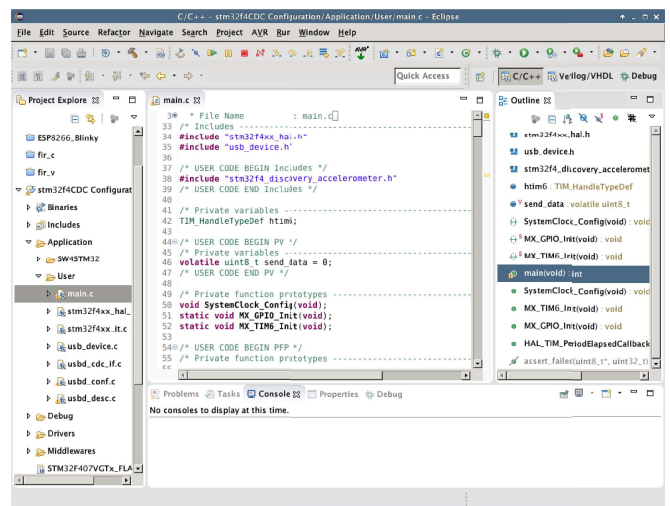


Figura 6. Eclipse como entorno de desarrollo para microcontroladores STM32.

IV-B1. Suite de desarrollo para STM32: Los microcontroladores STM32 son una familia de microcontroladores de 32 bits basados en procesadores ARM Cortex-M[19]. Ofrecen

un gran rendimiento, capacidades de tiempo real para control de procesos y procesamiento digital de señal y bajo consumo, manteniendo la simplicidad de uso mediante las librerías de abstracción del hardware que permiten programar fácilmente los microcontroladores en un lenguaje de alto nivel como C.

Se ha escogido esta suite como plataforma de desarrollo para la docencia de la asignatura de Sistemas Integrados del Grado en Ingeniería en Electrónica y Automática (Rama Industrial) de la Universidad de Extremadura. Los contenidos de la asignatura versan sobre concurrencia en microcontroladores y el control y ejecución de tareas de forma paralela. La ventaja de la plataforma, es que permite el uso de sistemas operativos de tiempo real (como es el caso de FreeRTOS y μ Clinux) para la docencia.

Al ser programados en lenguajes de alto nivel, favorecen el aprendizaje del alumno, pudiendo centrarse en tareas de planificación y control de la ejecución de tareas y aplicación de conceptos como Semáforos. Por tanto, resultan una elección adecuada para integrar en el laboratorio virtual.

ST ofrece placas de desarrollo a precios muy reducidos, ideales para las prácticas de la asignatura, y las herramientas integradas en dglab son de software libre, ya que se basa en el compilador de GCC para microcontroladores ARM.

dglab integra en Eclipse el compilador cruzado de GCC para microcontroladores ARM. Además utiliza OpenOCD como depurador para los microcontroladores.

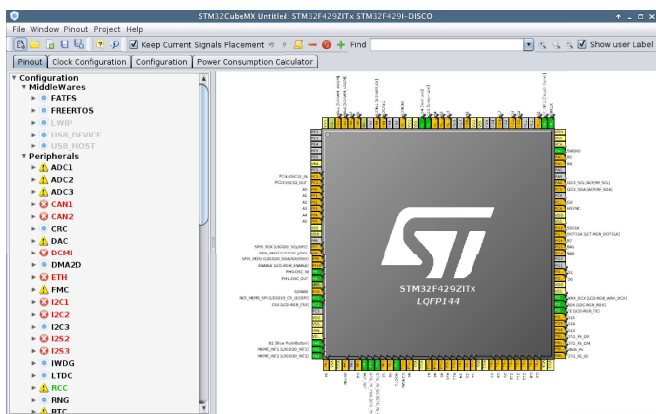


Figura 7. STM32CubeMX, Generación de ficheros de configuración.

Por último, la herramienta STM32CubeMX proporcionada por el fabricante se utiliza para la generación de código de inicialización. Una interfaz gráfica permite seleccionar los periféricos que se utilizarán, configurarlos y seleccionar algunas librerías de firmware para sistemas de archivos o implementación de clases USB entre otras cosas.

La generación automática de la configuración e inicialización de los periféricos suponen una gran ventaja para el alumno, puesto que permite al alumno centrar sus esfuerzos en el contenido de la asignatura, y abstraerse de los procesos de configuración e inicialización de periféricos, que no son objeto principal de la asignatura, y se presuponen competencias ya adquiridas en la asignatura Informática Industrial.

IV-B2. Compilador para AVR: Se ha integrado en Eclipse el compilador cruzado de GCC para dispositivos AVR. Los microcontroladores AVR son microcontroladores de 8 bits de la empresa Atmel, que son utilizados en las placas de desarrollo Arduino.

Se ha elegido esta plataforma y en especial el compilador con el objetivo de atraer a los alumnos que están acostumbrados por lo general al entorno de desarrollo de Arduino. Supone una gran ayuda en la enseñanza de los microcontroladores, puesto que parten de una plataforma conocida.

Para la programación de dispositivos se pueden utilizar diferentes programadores, que comprenden los dispositivos serie ICSP y programadores paralelos. La utilidad incluida para programación se llama *avrdude*. De nuevo es una aplicación de software libre y soportada por la comunidad, que puede programar la gran mayoría de los dispositivos AVR del mercado.

El plugin para eclipse que incluye las herramientas mencionadas anteriormente es *AVR Eclipse Plugin*[20].

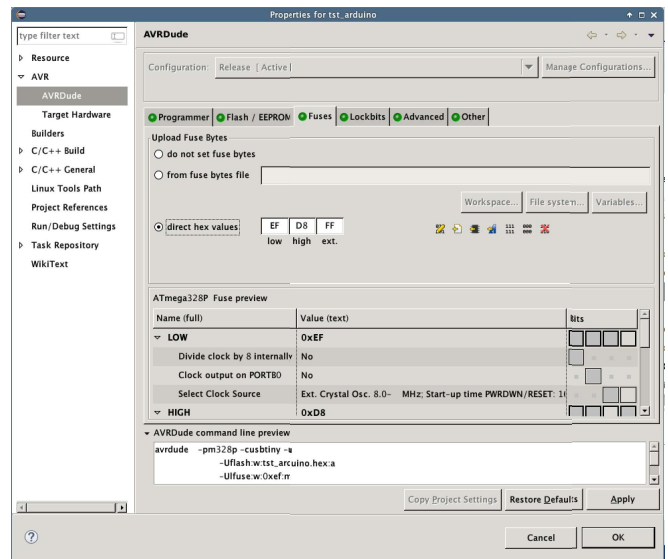


Figura 8. Configuración asistida de los Fuses de microcontrolador AVR.

IV-B3. SDK para el ESP8266: El ESP8266 es un SOC relativamente reciente, que integra tecnología WiFi con un procesador de aplicación de 32 bit. Con un coste muy reducido, es un chip muy popular para integrar tecnologías de comunicaciones inalámbricas de forma económica.

Entre sus características destacan:

- Comunicaciones SPI, UART y SDIO.
- Empaquetado QFN de 32 pines con alta densidad de integración.
- Interfaz I2S para transmisiones de datos de audio.
- Motores integrados WEP, TKIP, AES y WAPI.
- Conmutador RF integrado para comunicación WiFi.

Es capaz de comportarse tanto como un punto de acceso como un dispositivo conectado a otra red WiFi.

Se ha integrado en dglab para el desarrollo de Trabajos de Fin de Grado relacionados con los contenidos del área de la electrónica digital y la programación de microcontroladores.

Se incluye el SDK del fabricante, que contiene el compilador cruzado (de nuevo basado en GCC) y el programador para modificar la memoria flash (externa) del ESP8266 e instalar un nuevo firmware que permita el control completo del microprocesador (ya que inicialmente los módulos suelen tener instalados por defecto un firmware de comandos AT).

El SDK se integra en Eclipse mediante la gestión manual del fichero Makefile. De este modo, se tiene control preciso sobre las órdenes de terminal que se producen al construir el proyecto. Además, el alumno no necesita saber con detalle el funcionamiento del mismo, basta con modificar los parámetros base.

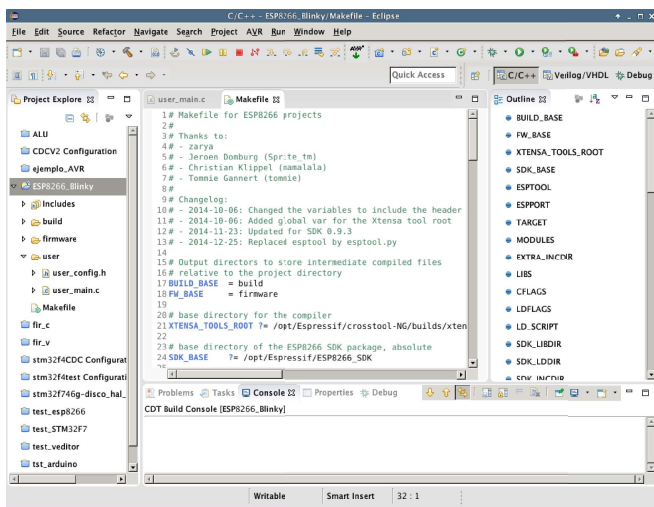


Figura 9. Eclipse y el SDK del ESP8266

IV-B4. Xilinx ISE Design Suite (Embedded Edition):

Opcionalmente se ha incluido en la máquina virtual la suite de desarrollo para FPGA de Xilinx. Si bien esta herramienta no es de software libre, se ha considerado adecuado integrarla puesto que a día de hoy no existen herramientas de software libre que puedan llevar a cabo el flujo de trabajo típico de las FPGA, desde la síntesis de diseños hasta la generación del bitstream que puede ser directamente cargado a FPGA en cuestión.

Sin embargo, a pesar de no ser un software libre, se encuentra conectado al servidor de licencias de la UNEX, y las licencias han sido donadas por Xilinx a través del programa universitario. De este modo, se evita cualquier tipo de costes a los alumnos y a la universidad, y se puede utilizar la suite completa para el desarrollo en FPGA.

Entre otros programas y utilidades, se integran los siguientes:

- **ISE.** Aplicación dedicada al diseño de módulos, su síntesis, mapeado y rutado en una tecnología de FPGA concreta.
- **XPS.** Aplicación dedicada a la integración de sistemas de procesamiento, incluyendo soft processors MicroBlaze.

- **SDK.** Entorno de desarrollo de software para los procesadores MicroBlaze basado en Eclipse.
- **Conjunto de IP's** que incluye periféricos y el microprocesador MicroBlaze.

La suite de diseño ISE se integra principalmente para proporcionar soporte a las placas de desarrollo de FPGA utilizadas en la asignatura de Diseño Digital del Grado en Ingeniería Electrónica y Automática (Rama Industrial).

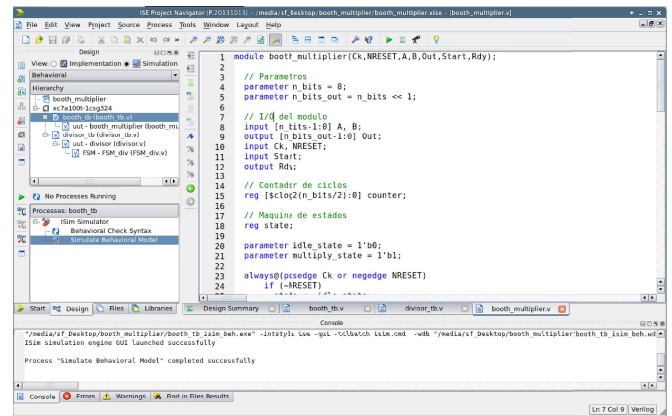


Figura 10. Xilinx ISE, Edición de módulos Verilog

IV-B5. Simulador de procesador MIPS: Para facilitar el aprendizaje sobre arquitecturas de microprocesadores MIPS se integra el simulador MARS de código ensamblador para estos procesadores. Entre sus características destacan la capacidad de ensamblar el código ensamblador a instrucciones en formato hexadecimal, y exportar el código para ser utilizado en el procesador en cuestión. Es también capaz de la simulación instrucción a instrucción, proporcionando una vista de los registros del microprocesador, y de los coprocesadores de control y punto flotante asociados al mismo.

Entre otros, tiene soporte para las mismas pseudoinstrucciones que soporta el microprocesador MIPS, y las ensambla en varias instrucciones básicas, que es como se simula finalmente. Está pensado para ser utilizado con el libro *Diseño y arquitectura de computadores*[21].

V. DOCUMENTACIÓN DE DGLAB

Además de proporcionar la máquina virtual, la página web de la Universidad de Extremadura aloja una Wiki[22] donde se encuentra la documentación de la máquina. Se muestran casos de usos de cada uno de los programas, con el objetivo de familiarizar al alumno en el entorno de trabajo de Eclipse para cada aplicación concreta.

Los ejemplos están enfocados a la demostración de las aplicaciones, no a la didáctica sobre los microprocesadores o diseños digitales concretos. Por ello, la mayoría incluyen ejemplos muy simples de seguir, simplemente para demostrar el flujo de trabajo y las operaciones requeridas en cada aplicación para los propósitos concretos de la misma.

Esta documentación no detalla todos los programas, sino que se trata de un punto de partida para los posibles escenarios

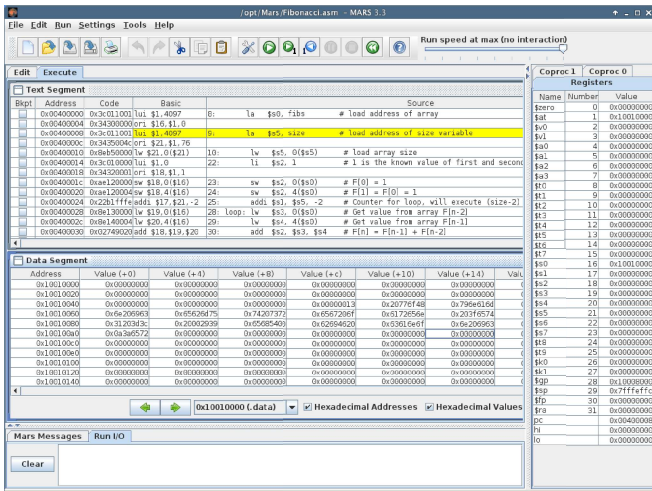


Figura 11. MARS, simulador de procesadores MIPS R2000/R4000.

en los que se encontrará el alumno. En caso de no tener claro algún detalle en particular, el alumno deberá dirigirse a la documentación específica del programa, donde se dispondrán todos los aspectos y características del mismo.

La idea de el laboratorio virtual, está inspirada en el antiguo laboratorio virtual de Fedora[23], personalizado para aportar las herramientas necesarias para las asignaturas a los alumnos.

La gran ventaja de su alojamiento en una Wiki online es la posibilidad de adaptar los contenidos a las posibles modificaciones que se efectuen sobre la máquina virtual y a las necesidades observadas por los alumnos en el aprendizaje de las herramientas.

VI. CONCLUSIÓN

dglab sirve como un entorno integrado para realizar las sesiones prácticas de las asignaturas de Electrónica Digital, Diseño Digital y Sistemas Integrados del Grado en Ingeniería Electrónica y Automática (Rama Industrial) de la Universidad de Extremadura, así como para el desarrollo de Trabajos de Fin de Grado relacionados con las materias. El alumno no necesita lidiar con los requisitos de instalación y aprendizaje sin tutorización de las herramientas que se requieren en el desarrollo de estas asignaturas. Los bajos requisitos técnicos de la máquina permite aprovechar los recursos existentes en los diferentes laboratorios, independientemente del sistema operativo de la maquina anfitrión.

REFERENCIAS

[1] Laboratorio de Electrónica Digital desarrollado por los autores del trabajo. <http://digital.unex.es/files>

[2] Canonical, Sistema operativo *Ubuntu*. <http://www.ubuntu.com/>

[3] IEEE Standard for Verilog Hardware Description Language, in IEEE Std 1364-2005 (Revision of IEEE Std 1364-2001), vol., no., pp.0_1-560, 2006 doi: 10.1109/IEEEESTD.2006.99495 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1620780&isnumber=33945>

[4] IEEE Standard VHDL Language Reference Manual, in IEEE Std 1076-2008 (Revision of IEEE Std 1076-2002), vol., no., pp.c1-626, Jan. 26 2009 doi: 10.1109/IEEEESTD.2009.4772740 keywords: hardware description languages; IEEE standard; VHDL language reference manual; VHDL hardware description language; electronic systems; formal notation; VHDL; computer languages; electronic systems; hardware; hardware design, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4772740&isnumber=4772739>

[5] *TkGate*, <http://www.tkgate.org>

[6] Eclipse Foundation, entorno de desarrollo integrado *Eclipse*, <https://eclipse.org>

[7] Visor de Formas de onda *GTKWave* <http://gtkwave.sourceforge.net>

[8] Herramienta de diseño de máquinas de estado *Qfsm*, <http://qfsm.sourceforge.net>

[9] STMicroelectronics, *STM32 Microcontrollers*. <http://www.st.com/web/en/catalog/mmc/FM141/SC1169>

[10] Atmel Corporation, Microcontroladores *AVR*, <http://www.atmel.com/products/microcontrollers/avr/default.aspx>

[11] Espressif Systems, SOC *ESP8266*, <http://espressif.com/en/products/esp8266/>

[12] Missouri State University, Mars MIPS Simulator, <http://courses.missouristate.edu/kenvollmar/mars/>

[13] Xilinx, suite de desarrollo ISE, <http://www.xilinx.com/products/design-tools/ise-design-suite.html>

[14] Oracle, *VirtualBox*, <https://www.virtualbox.org/>

[15] Entorno de escritorio *XFCE*. <http://www.xfce.org/>

[16] Plugin *VEditor* para la integración de lenguajes HDL en Eclipse. <http://sourceforge.net/projects/veditor/>

[17] *Icarus Verilog*, <http://iverilog.icarus.com>

[18] Simulador VHDL de software libre *GHDL*. <http://ghdl.free.fr>

[19] ARM Ltd. microprocesadores *ARM Cortex-M*. <http://www.arm.com/products/processors/cortex-m/index.php>

[20] *AVR Eclipse Plugin*. http://avr-eclipse.sourceforge.net/wiki/index.php/The_AVR_Eclipse_Plugin

[21] Patterson, D.A., Hennessy, J.L., Estructura y Diseño de Computadores, 4a ed., Reverte, 2011.

[22] Documentación para el Laboratorio Virtual desarrollada por los autores del trabajo. <http://digital.unex.es/wiki/doku.php?id=pub:dglab>. Usuario: pub Contraseña: digital.

[23] Red Hat Inc. *Fedora Electronic Lab*. <https://fedorahosted.org/fedora-electronic-lab/>