

Jlógica, un entorno de diseño de sistemas digitales

José A. Alvarez, V.G. Ruíz, J.F. Sanjuan, Javier Roca P.

Depto. Arquitectura de Computadores y Electrónica.
Universidad de Almería.
04120 Cañada de San Urbano.
jaberme@ual.es

Carlos A. Bermejo

Área de desarrollo – S. Informática.
Universidad de Almería.
04120 Cañada de San Urbano.
cbermejo@ual.es

Resumen

Jlógica es el resultado de una visión docente del lenguaje de programación Java. Con Jlógica se dispone de una herramienta de diseño y simulación de circuitos homogénea, intuitiva y sin restricciones, capaz de facilitar el aprendizaje en asignaturas tales como Estructura de Computadores, Tecnología de Computadores y Laboratorio de Arquitectura.

1. Introducción

Jlógica es una herramienta Java pensada para homogeneizar el entorno de trabajo en las prácticas de varias asignaturas, de esta manera módulos realizados para una determinada asignatura podrían ser reutilizables a medida que el alumno vaya adquiriendo conocimientos de mayor nivel con la consiguiente ventaja del conocimiento y experiencia práctica obtenida con el entorno de trabajo. Una de las principales ventajas de la herramienta es la independencia que otorga Java con respecto a la plataforma, así como la capacidad de integración con la red. La primera es importante ya que no restringe el sistema operativo usado en los equipos de los laboratorios. La segunda ventaja es que se hace extremadamente sencilla su utilización ya que una vez instalado en un servidor Web, el acceso puede realizarse tanto desde el laboratorio como desde cualquier punto conectado a la red.

Por otra parte, Jlógica dispone de un módulo generador de código Java a partir del diseño gráfico del circuito, el cual facilita el análisis del comportamiento del mismo mediante la modificación de parámetros básicos. Las asignaturas para las que se está desarrollando esta

aplicación son Tecnología de Computadores y Estructura de Computadores.

En este artículo se detallan las características de utilización de esta herramienta. En el apartado 2 se describe la interfaz y en el apartado 3 su utilización como entorno de simulación de circuitos. En el apartado 4 se detalla la utilización de esta herramienta como generador de código a partir del diseño gráfico de un circuito.

Además de haber optado por Java para la implementación por las razones antes aducidas, portabilidad e integración en la red, se pueden argumentar razones secundarias tales como que Java dispone de librerías de clases (Swing [4] y Awt [2]) que facilitan el desarrollo de la interfaz además de sobrados componentes visuales libres¹ y reutilizables [1].

2. Descripción de la interfaz.



Figura 1. Aspecto de la interfaz Jlógica.

¹ Libre ha de entenderse que el propietario del software cede el componente para su uso de manera desinteresada.

Jlógica cuenta como punto de entrada con una interfaz muy similar a la interfaz ofrecida por Outlook para relacionar la aplicación con las asignaturas que harán uso de ella y separar adecuadamente la funcionalidad de la aplicación. El panel de la izquierda dispone las áreas:

- Tecnología Computadores.
- Estructura de Computadores.
- Laboratorio de Arquitectura.

Dentro de cada una de estas áreas se encuentra la funcionalidad requerida para las asignaturas, sea el caso mostrado en Fig.1 en el que para la asignatura de Tecnología se dispone de opciones como *Simulador* que inicia la aplicación, *Enviar* para mandar al profesor resultados, etcétera. Pero la parte importante de la aplicación no es la interfaz que la relaciona con las asignaturas sino el propio simulador (Fig.2).

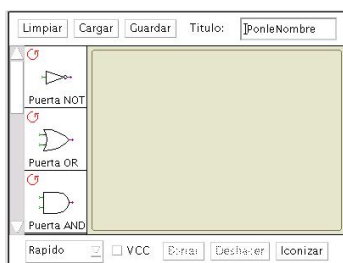


Figura 2. Aspecto del simulador.

El simulador dispone de un juego de puertas lógicas básicas (AND, OR, NOT) y de una serie de piezas de utilidad como conectores de entrada/salida y de empalme (Fig. 3), las cuales están dispuestas en la barra de puertas a la izquierda del área de trabajo Jlógica. Con estas puertas y piezas de utilidad el alumno puede comenzar a montar circuitos fácilmente. Además Jlógica cuenta con dos barras de herramientas (superior e inferior) que suman funcionalidad a la herramienta. La barra superior cuenta con funciones aplicables a todas las puertas del circuito, estas son:

- Limpiar: Elimina el diseño del circuito por completo.

- Cargar/ Guardar²: Exportación / importación del circuito a disco.
- Título: Nombra un diseño de circuito para usos posteriores.

En la barra inferior contamos con funciones aplicables a un subconjunto de puertas del circuito tales como:

- Borrar: Elimina las puertas seleccionadas así como sus conexiones con otras puertas.
- Deshacer: Restaura el diseño del circuito anterior a una modificación.
- Iconizar: Construye una nueva puerta con el diseño del circuito que esté en ese momento activo en el panel. Esta nueva puerta se añade con el nombre que se haya usado en la opción "Título" para que pueda ser usada como un componente en otros diseños.

Además de estas opciones aparecen otras dos que tienen que ver con la simulación en sí, estas son el desplegable en el que se puede elegir la velocidad con la que se verán la transición entre estados de una tabla de verdad y el checkbox³ *VCC* que es el que inicia la simulación haciendo las veces de fuente de alimentación.

3. Áreas de aplicación

Se entiende por áreas de aplicación a las asignaturas que pueden, en principio, hacer uso del simulador para adaptar parte de sus lecciones prácticas. Entre ellas se encuentran las que a continuación se enumeran.

3.1. Tecnología de computadores.

En esta asignatura Jlógica le permite al alumno comprobar de forma fácil cómo los resultados obtenidos tras una minimización correcta de funciones coinciden con las versiones canónicas de los mismos, bien expresada en su forma de tabla lógica o circuito combinacional.

² El circuito se copia en un fichero binario al que se le añade un identificador para que el simulador sólo abra los ficheros que él ha generado y no cualquier otro tipo de fichero.

³ Opción que se activa o desactiva marcando con un tick en el lugar habilitado.

La herramienta simula al panel-entrenador que con carácter general es utilizado por los alumnos en este tipo de prácticas, disponiendo de los elementos fundamentales integrados en el panel (interruptores, leds, conectores, etc.).

La disponibilidad de puertas lógicas, flip-flops u otro tipo de circuito lógico se obtiene a partir de la posible combinación de los elementos básicos (inversor, puerta AND y puerta OR) que se hayan integrados (Fig.2) con el desarrollo de la herramienta. De esta forma la tarea del alumno, ante el testeo de cualquier circuito propuesto, consiste en el ensamblaje apropiado de estos tres elementos hasta conseguir la función necesitada. Toda función diseñada anteriormente puede encapsularse como elemento de circuito y reutilizarse posteriormente en diseños más complejos, confeccionando de esta forma una librería tan extensa como se necesite. A continuación se describe un ejemplo.

La Fig.3 muestra cómo construir un circuito para implementar una determinada función.

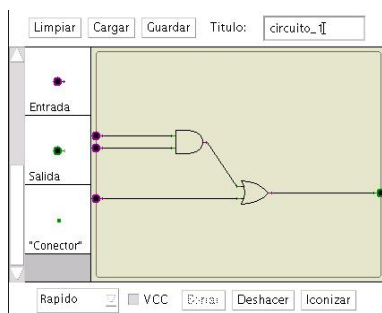


Figura 3. Diseño de circuito.

En Fig.4 se ve un ejemplo de cómo se encapsula para construir una puerta adicional llamada "circuito_1" con funcionalidad ampliada.

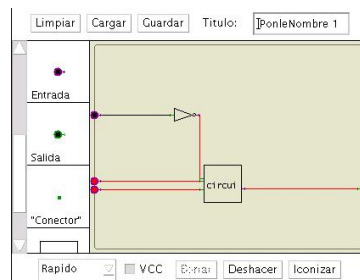


Figura 4. Encapsulación y reutilización.

En circuitos complejos donde el número de puertas sea elevado, existe la posibilidad de perder el diseño del circuito encapsulado que se está usando. La confusión es mayor si en lugar de un encapsulado se usan varios. Por tanto, Jlógica permite inspeccionar el contenido de los circuitos encapsulados sin más que seleccionar la pieza encapsulada y pulsar sobre el botón "Expandir"⁴, tal y como se puede ver en la Fig. 5.

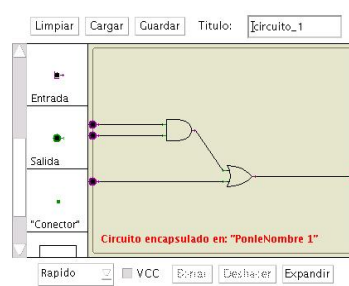


Figura 5. Contenido del encapsulado utilizado en el montaje.

3.2. Estructura de computadores.

El enfoque que se le otorga a esta asignatura en nuestro actual plan de estudios es el de iniciar al alumno en el conocimiento de los bloques fundamentales constitutivos de un computador.

⁴ La opción expandir sólo se hace visible cuando existen componentes compuestos en el circuito.

El alumno al finalizar la asignatura ha debido adquirir el conocimiento de la metodología de diseño a nivel de registros así como la estructura y organización de una unidad central de proceso. Así pues, el aprendizaje realizado con módulos lógicos elementales en la asignatura Tecnología de Computadores son aplicados en esta asignatura para constituir bloques más complejos como ALU's o el diseño completo de una unidad central de proceso sencilla.

Las características de Jlógica explicadas anteriormente facilitan la consecución de estos objetivos sin más que sustituir el conjunto de piezas básicas usadas en la asignatura Tecnología de Computadores por un nuevo conjunto de piezas necesarias para esta asignatura como pueden ser una ALU, bancos de memoria, bancos de registros, multiplexores, etcétera.

4. Generación de código.

Uno de los aspectos prácticos tratados en Laboratorio de Arquitectura de Computadores es la simulación computacional de circuitos lógicos [3], mediante la que el alumno experimenta el testeo de bloques computacionales constitutivos de una arquitectura analizando el comportamiento ante distintos patrones de entrada y modificación de parámetros característicos.

Jlógica facilita el análisis anterior mediante la generación de código (Fig.6) a partir del diseño gráfico del bloque a estudiar. El modelo en esta versión se genera en código Java, si bien se pretende ampliar a otros lenguajes tal capacidad.

5. Conclusiones.

Jlógica goza de las capacidades propias del lenguaje Java como son la independencia de la plataforma lo que le confiere gran portabilidad y la hace una herramienta muy flexible y poco restrictiva, integrable en Internet. Jlógica se integra en páginas Web sin dificultad (applet). Otra ventaja de Jlógica es el carácter interdisciplinario que adquiere al poder utilizarse en diferentes asignaturas.

El desarrollo de Jlogica es el fundamento de una herramienta con la que se pretende en un futuro inmediato alcanzar niveles de complejidad de una arquitectura completa, abordando tanto el diseño del camino de datos como el de la unidad de control.

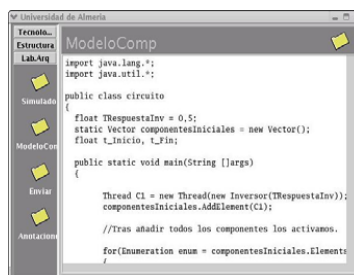


Figura 6. Ejemplo de generación de código para un inversor.

Referencias

- [1] Duguay, Claude. Revista electrónica JavaPro. <http://www.fawcette.com/archives/magazines/javapro>
- [2] Knudsen, Jonathan. Java 2D Graphics. First Edition. ISBN 1.56592-484-3. Ed. O'Reilly.
- [3] Ruiz, V.G. *Diseño de Simuladores Digitales usando SDL++*. IV Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica (TAEE). Barcelona, Spain. Septiembre. 2000.
- [4] Walrath, Kathy y Campione, Mary. The JFC Swing Tutorial, a guide to construct GUIs. ISBN 0-201-43321-4. Ed. Addison-Wesley.