

# Herramientas de Diseño Lógico y Simulación para Ayuda al Estudio en las enseñanzas de Estructura de Computadores

**Lourdes Peñalver Herrero, Ana Pont Sanjuan,  
José Antonio Gil**  
email :{lourdes, apont, jagil} @disca.upv.es  
Universidad Politécnica de Valencia.  
Departamento de Ingeniería de Sistemas  
Computadores y Automática

*En este artículo se presentan las experiencias adquiridas en el desarrollo de un paquete de diseño digital de dominio público dirigido a la realización de prácticas de las asignaturas de Estructuras de Computadores para las titulaciones de informática o equivalentes.*

*En el desarrollo del paquete se ha seguido una metodología de trabajo encaminada a que los alumnos de proyectos adquieran una experiencia útil en su desarrollo profesional futuro.*

## 1. Introducción y Motivación

**L**AS enseñanzas Técnicas se caracterizan principalmente por su alto contenido en prácticas, dirigidas a que el alumno pueda tener una formación integral. En este grupo de enseñanzas técnicas destacamos los estudios de informática de manera muy especial por su rápida evolución tanto tecnológica como de aplicación. Para realizar dichas prácticas en las escuelas cuando no está disponible el material por su elevado coste, o de forma complementaria se recurre a programas de simulación que permitan al alumno realizar las mismas adaptándose a los rápidos cambios que se suceden en estos campos. Dichos programas, aunque disponibles en las escuelas, suelen ser excesivamente caros para que el alumno los pueda adquirir. Para solventar este problema, cada día más sobre todo en las universidades del extranjero existen programas de dominio público que el alumno se puede llevar a casa para que le ayude en el estudio. En esta línea median-

te proyectos fin de carrera se ha realizado un programa de simulación lógica que permita al alumno realizar las prácticas del grupo de asignaturas formado por Fundamentos de Computadores, FCO, Estructura de Computadores I. EC1, y Estructura de Computadores II, EC2, correspondientes a los tres primeros cuatrimestres de las titulaciones de Ingeniería Técnica en Informática de Sistemas y de Gestión de la E. U. de Informática de Valencia.

Dicho programa se ha realizado siguiendo una metodología de programación orientada a objetos, y cubre las necesidades de las prácticas de las asignaturas citadas. Debido a la complejidad del problema requiere ser realizado por un grupo de trabajo, en nuestro caso formado por cuatro alumnos de Proyecto Fin de carrera. Esta ha sido principalmente una de las razones por la que se ha elegido la metodología de diseño orientado a objetos ya que permite desglosar el trabajo en módulos independientes.



La asignatura de FCO centra su estudio principalmente en el diseño lógico, cuyo nivel de abstracción más bajo es el concepto de puerta que coincide con el de programa, y en el estudio de lenguaje ensamblador. Para este último ya se dispone de una herramienta de dominio público de simulación del ensamblador del procesador R2000 de MIPS. Las prácticas correspondientes al diseño lógico se realizan mediante la utilización de un entrenador lógico, por lo que para esta materia el programa sólo se utiliza como una herramienta complementaria que sirva al alumno para ampliar su estudio o realizar ejercicios en casa.

En las asignaturas de EC1 y EC2 se estudia la estructura del computador, en la primera la parte correspondiente a la Unidad Central de Proceso, o Procesador, y en la segunda la Memoria y La unidad de Entrada Salida. Las prácticas relativas a EC1, se basan principalmente en el diseño de una Unidad Central de Proceso (UCP), sencilla, o circuitos pertenecientes a la misma utilizando elementos lógicos estudiados en la asignatura de FCO. Asimismo en la asignatura de EC2 las prácticas se centrarían en la realización de circuitos que permitan dotar la UCP ejemplo desarrollada en EC1 y asignar a ésta una unidad de memoria y de entrada/salida, además de alguna práctica encaminada a la utilización de la entrada/salida de un procesador real, como un PC. Para estas prácticas es necesario disponer de una herramienta para la realización de las mismas y trabajos complementarios que se puedan plantear.

## 2. Descripción del programa

El programa se ha planteado, como se ha comentado anteriormente siguiendo una metodología de programación orientada a objetos. Debido a la complejidad del mismo esta metodología permite hacer un estudio de cuáles son los elementos que deben componer el programa final y a partir de ahí distribuir el trabajo, en nuestro caso en cuatro trabajos que presentará cada alumno como proyecto final de carrera. Además no sólo se han propuesto como trabajos independientes sino que se consideró desde el principio que era interesante que los alumnos trabajasen en grupo, situación común en el ejercicio de la profesión tanto fuera como dentro de la universidad. Por lo tanto se planteaba como un ejercicio cuya meta era realizar el proyecto final de carrera y una experiencia de trabajo en grupo, para llevar esto a cabo la tarea del profesor, que dirige los proyectos dentro de este grupo, ha sido la de coordinador del mismo.

A la hora de plantear el trabajo se consideró intere-

sante dividirlo en varios apartados diferentes :

- una interfaz amigable con el usuario, que permitiera un uso sencillo e intuitivo del programa. Esta interfaz sería de tipo ventanas y menus desplegables con una ayuda en línea de los elementos disponibles.
- un elemento que permitiera el diseño de circuitos, para reducir complejidad se consideró interesante distinguir entre circuitos combinacionales y secuenciales, aunque los dos trabajos debían estar íntimamente relacionados entre sí.
- y por un elemento que permitiera la simulación de los circuitos diseñados.

El comienzo del trabajo se consideró oportuno abordar todo al mismo tiempo en trabajos diferentes pero coordinados, otra alternativa hubiera sido realizar una primera aproximación que se fuera depurando en fases sucesivas. La primera opción nos pareció más interesante aunque arriesgada por lo siguiente : plantear un problema desde el principio nos da una visión general del mismo difícil de obtener si no se estudia en su completitud en todas las fases de desarrollo, ya que muchas veces es necesario una vuelta atrás y replantear algunas cosas, como así ha sucedido en varias ocasiones. Es arriesgado porque todos sabemos el grado de compromiso del alumno en los proyectos final de carrera, y un abandono a mitad de trabajo, o una mala coordinación puede echar al traste todo el proyecto. Para solventar este último inconveniente se hizo una rigurosa selección de los proyectandos. Se desarrollo un trabajo intensivo, al principio, del profesor con los alumnos para que permitiera una puesta en común de las ideas a la vez que se evaluaba las dificultades de coordinación que pudieran surgir entre los mismos, se sentaron las bases de las distintas partes y se establecía un calendario de tareas y un plan de trabajo a seguir.

Teniendo todo lo anterior en cuenta el trabajo se dividió en cuatro proyectos final de carrera, cada uno de los cuales encargado de uno de los módulos siguientes :-

- La interfaz de usuario
- Diseño de circuitos combinacionales.
- Diseño de circuitos secuenciales.
- Simulación de circuitos.

Otra cuestión importante era ¿qué tipo de programa queríamos ? un programa de simulación y diseño profesional con una librería de los circuitos integrados existentes en el mercado como puede ser el **ORCAD**, o un programa educacional orientado a la simulación como el **CASCAD**, programa actualmente en uso en las prácticas consideradas. La elección fue claramente hacia la segunda alternativa ya que nues-

tro objetivo principal es la de disponer de una herramienta de simulación que permita a los alumnos realizar las prácticas, pero haciendo un diseño que permita añadir en un futuro librerías actualizables que doten al programa de una entidad suficiente.

Una vez decidido esto debíamos definir teniendo en cuenta los objetivos de las asignaturas, el tipo de prácticas y de alumno las características del programa, como :

- elemento básico de diseño,
- utilidades del diseño como la capacidad de agrupar circuitos y de utilizarlos como elementos para realizar circuitos más complejos o añadirlos a la librería de circuitos definidos.
- utilidades de simulación que permitieran al alumno comprobar el funcionamiento de los circuitos.
- utilidades de diseño que permitieran mover y borrar elementos fácilmente, etc.

En suma, todos aquellos elementos que permitieran al alumno realizar como fin último el diseño de un computador a partir de elementos sencillos.

Una vez definido todo esto, había que elegir el lenguaje y el entorno de trabajo del programa. El primero, teniendo en cuenta que la idea era realizar un programa siguiendo la metodología de programación orientada a objetos, debía ser un programa que tuviera esta definición y de uso lo suficientemente extendido que facilitara mejoras posteriores, por lo que se utilizó el C++ para de los módulos de diseño de circuitos y simulación. La interfaz gráfica como era una parte íntimamente relacionada con el entorno de trabajo se ha de considerar junto con éste. Como entorno se decidió utilizar, windows o windows95 ya que las herramientas que disponemos para realizar las prácticas son PC's que trabajan con estos entornos. Por lo tanto la interfaz se debía realizar con programas que facilitaran el uso de esta interfaz : VisualBasic, Visual C++, Delphi. Se optó por esta última por la facilidad de uso y su definición claramente orientada a objetos.

Con todo esto se planteó el programa siguiendo los siguientes pasos :

En primer lugar había que diseñar a grosso modo

cada una de los módulos del programa, y definir la interacción existente entre ellos.

Para realizar esta tarea se partió de la idea de tener un objeto común a todos los módulos, que era el circuito, con una serie de características como:

- número de entradas,
- número de salidas,
- representación gráfica,
- tiempo de retardo,

Además debía soportar operaciones como : obtener la salida en función de las entradas, trabajo del módulo de simulación, dibujar el circuito con sus entradas, y salidas, objetivo de la interfaz etc.

Todo este trabajo se fue refinando en sucesivas reuniones, hasta que el grupo de alumnos fue capaz de trabajar por sí sólo y cuyo resultado pasamos a describir :

En un esquema general los diferentes módulos se muestran en el esquema de la figura

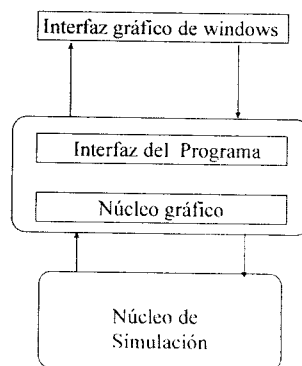


Figura 1 Esquema general de los diferentes módulos desarrollados.

### 2.1. Interfaz gráfica

Como se ha comentado anteriormente se ha realizado utilizando Delphi sobre windows, con un esquema familiar para el alumno como se muestra en la Figura 2. En esta figura se ha activado la pestaña de las puertas lógicas, si se selecciona la pestaña de circuitos combinatoriales; se muestran algunos circuitos básicos como pueden ser decodificadores, codificadores, etc.

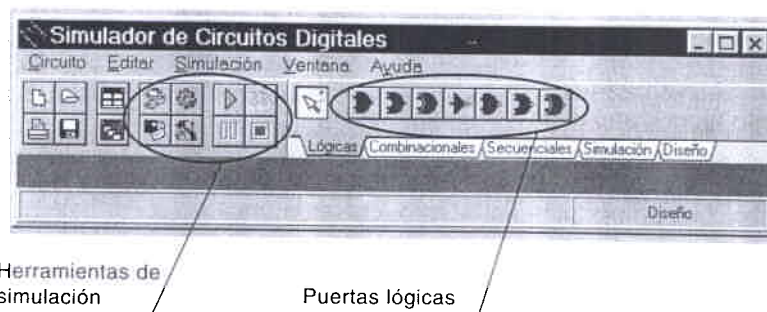
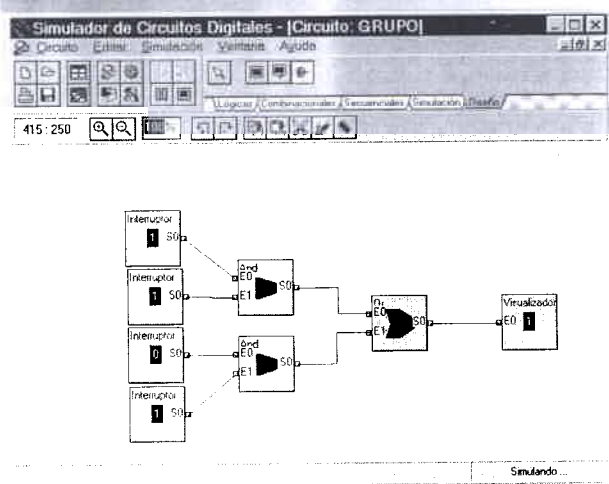


Figura 2 : Ventana principal, donde se muestran los principales iconos de las herramientas, y las puertas básicas.

## 2.2 Circuitos combinacionales.

En este módulo se utiliza como elemento básico las puertas lógicas que permiten definir cualquier tipo de circuito combinacional. A partir de las mismas se han implementado elementos útiles para la creación de circuitos combinacionales más complejos como pueden ser: decodificadores, codificadores, multiplexores etc. Para todos estos elementos se pueden definir el número de entradas y salidas, así como los retardos asociados.

*Figura 3. Aquí se muestra un ejemplo de un circuito combinacional que implementa la función  $ab+cd$ , donde a cada entrada del circuito se ha conectado un interruptor que permite cambiar el valor de las entradas, y a la salida un visualizador.*

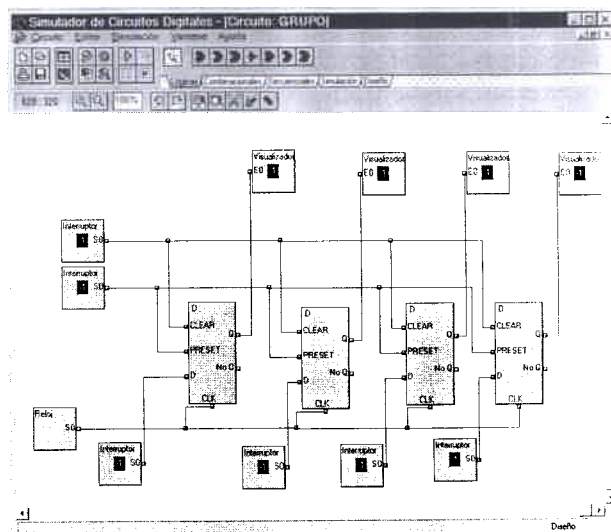


## 2.3 Circuitos secuenciales

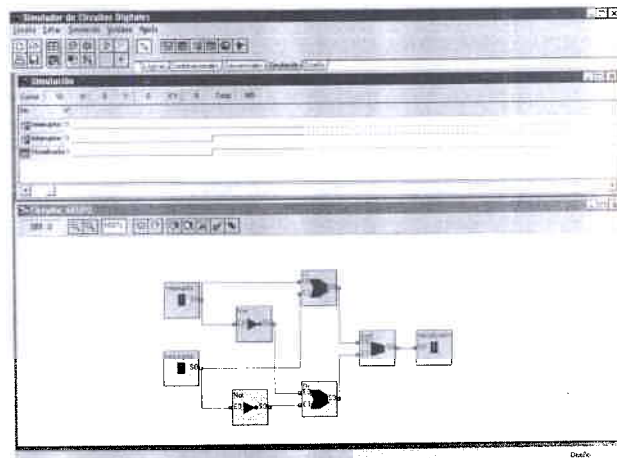
De forma similar se han definido elementos básicos secuenciales como son los biestables, a partir de los cuales se pueden construir cualquier circuito básico como son registros y contadores útiles para la realización de las unidades funcionales del computador. así mismo dispone también, ya definida, de una unidad de memoria a la que se le puede definir el tamaño de palabra y el de la dirección. En la Figura 4 se muestra un ejemplo de un registro de cuatro bits.

## 2.4 Módulo de simulación

Es el encargado de implementar todas las funciones que permiten que los circuitos funcionen como corresponde. Esta simulación se puede observar a partir del valor de las salidas del circuito como se muestra en Figura 3 y Figura 4, o como una gráfica de tiempos como se muestra aquí:



*Figura 4. Registro de cuatro bits con entrada paralela. En esta figura todos los biestables disponen de entradas asincrónicas conectadas en todos los registros al mismo valor, así como la entrada de reloj. A las entradas D de los biestables se le ha conectado un interruptor que permiten cambiar el valor de la entrada, y a la salida un visualizador para observar el estado del registro.*



*Figura 5. Se muestra una imagen con dos ventanas una con un circuito ejemplo y otra con la simulación del valor de las entradas y salidas.*

## 3. Conclusiones

Se han presentado las experiencias adquiridas en el desarrollo de una herramienta de diseño lógico, que permite la realización de las prácticas correspondientes a las asignaturas de Fundamentos y Es-

estructura de Computadores, mediante proyectos final de carrera.

De esta experiencia cabe destacar el hecho de que los alumnos que han realizado el proyecto han trabajado con planteamientos profesionales en la realización de un programa disponible de forma gratuita para todos los alumnos y profesores que así lo soliciten.

La utilización de la metodología de programación orientada a objetos así como el uso de lenguajes muy extendidos permitirá la ampliación o modificación del mismo fácilmente.

Además como trabajos futura se plantea entre otros añadir una librería de circuitos disponibles en el mercado.

#### 4. Referencias

[1] Logic Works , Interactive Circuit design software. Capilano Computing Systems, LTD.

[2] Bjarne Stroustrup. El C++ Lenguaje de Programación. Addison Wesley/Díaz de Santos. Segunda Edición, 1993.

[3] Roger Pressman Ingeniería del Software. Un enfoque práctico. McGraw-Hill. Tercera Edición. 1995. ■

