

SISTEMA DE SIMULACIÓN PARA EL DISEÑO LÓGICO Y DE COMPUTADORES.

Juan Pérez Rodríguez¹, Benedicto Gómez Díaz¹, Josep M^a Ribes i Bonet¹, Francesc Escudero i Costa¹

¹*Enginyeria i Arquitectura La Salle*
e-mail: juanp@salleURL.edu, se05050@salleURL.edu
jmribes@salleURL.edu, sisco@salleURL.edu

RESUMEN: En el presente artículo se presenta el programa de apoyo a la docencia que ha sido desarrollado por profesores de la asignatura de Introducción a los Ordenadores. Dicho programa permite la simulación de circuitos lógicos, ya sean básicos o de gran complejidad. La finalidad del simulador es la de ayudar a comprender el funcionamiento de las diferentes arquitecturas de computadores que se estudian en la asignatura y para ello se introduce en la parte práctica como herramienta de trabajo habitual. Las diferentes características del simulador permiten el diseño de arquitecturas nuevas, la simulación lógica de los diseños y el tratamiento de los bancos de memoria.

1.- INTRODUCCIÓN.

Desde la asignatura de Introducción a los Ordenadores, que se encuentra presente en el primer curso de todas las carreras de ingeniería de la Escuela Universitaria La Salle, se ha querido dar un alto grado de importancia a la realización de prácticas en el marco de la asignatura. Para ello se decidió desarrollar un software capaz de simular los circuitos lógicos que se enseñan a principios de curso con el Álgebra de Boole [1], los elementos de memoria que se explican a medio curso y las estructuras de un computador[3], que se explican a finales de curso.

La razón principal para desarrollar este software de simulación es la de poder contar con una herramienta versátil en la parte práctica de la asignatura que permita al alumno comprender en profundidad aquellos conceptos abstractos que ve en la teoría y que no es capaz de imaginarse. Para conseguir este propósito es cierto que existen paquetes de software en el mercado, pero la mayoría quedan limitados en cuanto a la posibilidad de modificar la arquitectura a estudiar o la forma de ver la evolución del sistema o, simplemente, son paquetes muy generales donde esta simulación es una parte más, poco desarrollada. Con un software propio se puede crear la herramienta adecuada al temario explicado y se dispone de la posibilidad de modificar en un futuro, con cierta facilidad.

Es muy fácil que esta idea de un paquete de simulación crezca en expectativas hasta intentar abarcar temarios o aspectos de asignaturas posteriores a la de Introducción a los Ordenadores. Y es por ello, que en su diseño se tiene en cuenta posibles ampliaciones y simulación de nuevas arquitecturas, así como la incorporación de un juego de instrucciones. Sin embargo, este artículo

tratará de la primera versión del simulador, en la que muchas cosas no estarán disponibles. Sólo cubrirá las necesidades de un primer curso de carrera universitaria.

No debe olvidarse el marco en que se realiza este paquete de simulación; se trata de una ingeniería común para informática, telecomunicaciones, electrónica y otras especialidades. No sería lógico dejar de lado, en este proyecto, las necesidades específicas de cada especialidad ya que el simulador estaría sentenciado al fracaso. Por ello, se intentan integrar características de simplificación de funciones lógicas (propias de una ingeniería electrónica) con métodos informáticos como Quine-McCluskey[4], la introducción de funciones por tablas de Karnaugh[5], la creación de componentes (muy útil en Informática para el diseño de nuevas arquitecturas), la simulación de memorias o el control del retraso sufrido por las señales (muy interesante en el campo de las comunicaciones). En resumen, un conjunto de características que en mayor o menor grado acabarán siendo útiles en todas las especialidades.

A lo largo de este artículo se describirán los datos más relevantes del simulador y se intentará dar una explicación de cómo se llevó a cabo el diseño del mismo. El siguiente apartado dará una idea del alcance del proyecto al exponer sus características principales y cuales son las que se han contemplado para una futura versión. Tras las características se exponen los criterios utilizados para seleccionar como debía ser el simulador y se cierra la ponencia con las conclusiones de la utilización del software de simulación, valorando el resultado final obtenido y su introducción en la asignatura.

2.- CARACTERÍSTICAS DEL SIMULADOR.

a) Condicionantes.

Las características del sistema en su primera versión son limitadas debido a una cuestión de tiempo, ya que el simulador debe estar funcionando a partir de setiembre de este mismo año para poder realizar pruebas de simulación. Ello obliga a que la estructura del programa este muy bien pensada para poder realizar ampliaciones en el futuro.

Otro condicionante, que no deja de ser una característica, es el entorno de trabajo para el alumno. Se debe pensar en un programa de fácil manejo y de una adaptación muy rápida. De esta manera se intenta evitar el rechazo por parte del alumno.

b) Características actuales.

Para hablar de las características del simulador es necesario recordar que se pensó para alumnos de primero de carrera y que, por tanto, debe encararse a un tipo de usuario que no esta acostumbrado a utilizar este tipo de simuladores. Como se ha dicho anteriormente, ello obliga a que la característica principal sea un entorno fácil de utilizar y muy intuitivo, con ordenes rápidas y sencillas como poner componente, realizar conexiones, simular, etc.

Centrándonos en el alcance de lo que puede hacer el programa, se puede empezar por las características de un diseño lógico. Para ello permite realizar diseños utilizando componentes existentes en las librerías básicas o crear componentes con la descripción de pines y el comportamiento que tiene. Permite crear nuevas librerías, incluir componentes que provienen de diseños ya realizados (encapsulando el diseño en un componente), crear componentes que responden a tablas de verdad o tablas de Karnaugh y cuya función lógica se obtiene a partir de la simplificación lógica a dos niveles mediante el método de Quine-McCluskey[4]. En estos

diseños se permite la introducción de fuentes de señal (onda cuadrada, continua o una forma de onda digital concreta, cuyos valores se establecen manualmente o con un fichero) y puntos de control que más adelante aparecerán en la simulación para poder estudiar su evolución en el tiempo. De la misma manera se podrían definir las entradas y salidas de un circuito y, de esta manera, poderlo convertir en un componente que se utilizará en posteriores diseños.

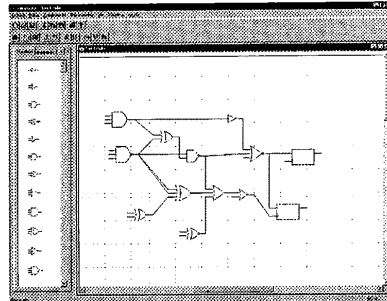
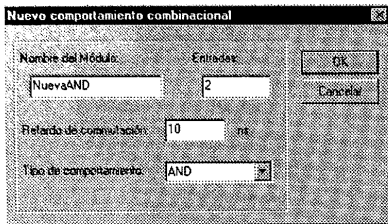


Figura 1. A. Proceso de creación de un nuevo componente a partir de la definición de los pines y su comportamiento. B. Ejemplo de un diseño lógico con puertas elementales y biestables.

Por otro lado, todo circuito o diseño que se realiza puede ser simulado y estudiado a partir de la evolución de las señales y de los valores que adquiere. Para realizar esta parte de forma visible se dispone de la pantalla de simulación en la que se muestran las señales escogidas en el diseño, con diferentes zooms y marcadores temporales. Tal pantalla se muestra en la figura 2.

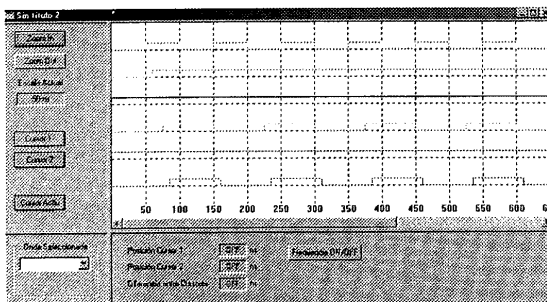


Figura 2. Pantalla con la simulación de las señales escogidas en el diseño.

El siguiente paso lleva a hablar de las características de los elementos de memorización. Básicamente, se dispone de los diferentes tipos de biestables y de las memorias estándares de mercado. Otros elementos intermedios, como pueden ser los registros se pueden construir a partir de estas librerías mínimas (un registro será una agrupación de biestables). En ellos se pueden definir los retrasos que sufren las señales y algunas propiedades más. Lo importante es que a nivel de memorias se permite rellenar esta con los valores de un fichero, con los valores introducidos manualmente o con los valores que introduzca el circuito. Sea de la forma que sea, se ofrece una herramienta para observar la evolución de los valores introducidos. Y esta característica es muy importante de cara a la simulación de arquitecturas de computadores.

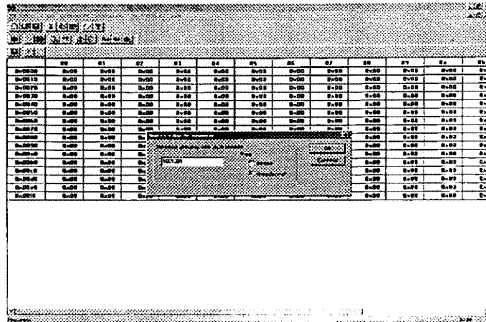


Figura 3. Pantalla en la que se observa el trabajo con la memoria y los valores que se introducen.

Las características que más hacen de este simulador una herramienta para la enseñanza son las que permiten la simulación y el diseño de pequeños núcleos de computadores. Para ello, se partirá de que será necesario disponer de una ALU, de un banco de registros y de una unidad de control (UC) que estén creadas para tal propósito.

Siendo así, el programa permite diseñar la UC de tipo microcableada a partir de la maquina de estados a la que tiene que responder. Permite la definición de las señales de entrada, de las señales de salida, los ciclos de reloj que puede ocupar y los opcodes que debe reconocer. Por ahora, el programa no permite el diseño de una UC microprogramada. Respecto a la ALU solo se permite la configuración de las instrucciones que reconoce, las operaciones a realizar y los tiempos que tarda en realizar dichas operaciones.

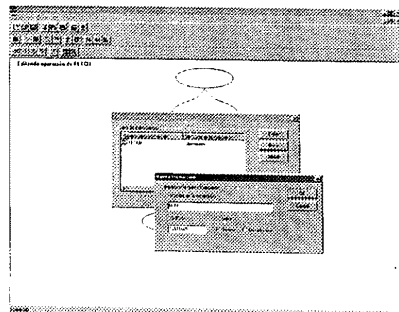
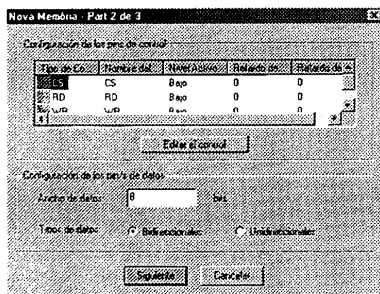


Figura 4. Detalle del diseño de una Unidad de Control, con su máquina de estados y el conjunto de señales de entrada y de salida.

De toda esta última parte, cabe destacar que el programa permite unir todos los elementos para que el diseño funcione como el núcleo de un pequeño computador, en el que se puede observar la evolución de los valores de los registros importantes, de los datos que pasan por el datapath y de la evolución de la memoria. Y se tiene acceso a toda esta información porque es el usuario el que ha creado la máquina que esta simulando y, por tanto, esta viendo el circuito.

c) Características futuras.

Entre las características futuras destacan una interficie capaz de entender VHDL para pequeños diseños o circuitos lógicos. Esta interficie será capaz de leer datos y de escribir datos en este lenguaje. La utilidad más importante será llevar los diseños a otros programas a través de un formato estándar.

Se intentará mejorar toda la parte de diseños de unidades aritmético-lógicas, para conseguir una mayor flexibilidad en los parámetros que se pueden configurar y en las operaciones a realizar. Tampoco se descarta la posibilidad de implementar ALUs donde se definan los procesos de cálculo, y por tanto, los tiempos que tardará por cada operación en función de la forma de calcular los datos.

En la misma línea, se debe buscar el diseño de una unidad de control de tipo microprogramada. De esta manera, la parte de diseño de arquitecturas de computadores quedaría bastante completa, teniendo en cuenta que se buscan pequeñas arquitecturas de carácter docente y no arquitecturas excesivamente complicadas.

Debe mejorarse la forma en que se ven los resultados de la simulación, no a nivel lógico de las señales, sino para las memorias, buses y grandes cantidades de datos.

Finalmente, destacar la posibilidad futura de cargar un programa en código máquina, obtenido por el proceso de compilación de un software estándar de mercado, sobre la memoria del sistema para que sea ejecutado sobre la arquitectura diseñada, siempre y cuando coincida el juego de instrucciones y los opcodes de ambas partes (programa y arquitectura). Actualmente, las instrucciones se introducen manualmente y se simula el comportamiento de dicha instrucción.

3.- ELECCIÓN DE CRITERIOS.

Los criterios para elegir las características del programa final se basan en los criterios de la asignatura impulsora y en las necesidades que esta tiene de cara a poder afianzar los conocimientos teóricos adquiridos.

Es fácil observar que para entender un circuito lógico básico no es necesario tener un potente simulador. Sin embargo, cuando se pide al alumno entender el movimiento de datos dentro de un computador, su evolución temporal o determinar donde se encuentran los problemas de una arquitectura concreta, se necesita poder trabajar con la máquina, ver lo que hace y poder modificar los procedimientos a seguir. Esto es algo impensable con una máquina real, debe utilizarse una simulación que sea lo más parecida a la realidad y suficientemente versátil como para permitir cambios.

Existen dos criterios principales que se han seguido para diseñar el simulador y que responden a las finalidades de la asignatura, diseño lógico a bajo nivel y con un alto grado de escalabilidad y simulación de computador con acceso a todos los puntos del sistema.

El primer criterio lleva a realizar un simulador que permita la realización de pequeños circuitos, su simulación y su posterior integración como componente discreto en grandes diseños.

El segundo obliga a la realización de los componentes de un computador por partes para tener acceso a todos los puntos. Por tanto se debe permitir el diseño de una unidad de control, de una ALU y de todo el conjunto de registros e interconexiones que forman la arquitectura.

Ello implica definir los criterios para diseñar este nuevo tipo de elementos tan específicos. El principal es tener una herramienta potente pero a la vez que sea ágil, para que el trabajar con ella no sea una cosa pesada.

4.- CONCLUSIONES.

Por ahora es demasiado pronto para sacar conclusiones acerca del grado de utilidad del simulador, ya que empezará a aplicarse con los alumnos a partir del próximo curso. Sin embargo, la experiencia con otros simuladores y las impresiones recogidas de los alumnos, hacen pensar que esta herramienta será de gran utilidad.

También se debe pensar que el simulador será entendido como un complemento a la teoría que se explica en clase. Teoría que muchas veces es abstracta y no se acaba de comprender porque no es fácil de imaginar. Con esta herramienta se resuelven muchos de estos problemas y permite un "margen de maniobra" para adaptarse a aquello que se explica en la asignatura.

En cuanto al resultado de la idea original se puede concluir que la primera versión responde a casi un 90 % de lo que se pretendía en un principio. Sin embargo, si tuviera que ser evaluada actualmente, con las nuevas ideas que se han propuesto, seguramente respondería a un 40-50% de lo que se pretende. Ello indica que existirán futuras versiones mejoradas y que poco a poco será una buena herramienta de trabajo para los alumnos.

5.- REFERENCIAS.

- [1]. 'Principios Digitales'. Tercera Edición. Roger L. Tokheim. Mc Graw-Hill. 1995.
- [2]. 'Introducción al diseño lógico digital'. Primera Edición. John P. Hayes. Addison-Wesley Iberoamericana. 1996.
- [3]. 'Computer System Architecture'. Tercera Edición. M. Morris Mano. Prentice Hall. 1993.
- [4]. 'Prime implicants by Quine-McCluskey Tabulational Method'. <http://www.capitol-college.edu/academics/ugrad/classes/oldho/el304/quine.htm>
- [5]. 'The Map Method for Synthesis of Combinational Logic Circuits'. M. Karnaugh. Transactions of the AIEE, Part I, vol. 72, págs. 593-599. 1953.