

# Simulador de dispositivos de entrada/salida programables

Manuel Prieto, Antonio J. Vicente, José. A. Vargas

Departamento de Automática  
Universidad de Alcalá  
28871 Alcalá de Henares  
e-mail: mpm@aut.uah.es

## Resumen

La ponencia presenta un entorno de simulación de la entrada/salida de sistemas basados en la familia i80x86. Está enfocados principalmente a servir como apoyo a los alumnos de la Universidad de Alcalá en las asignaturas de Sistemas Electrónicos Digitales y Laboratorio de Sistemas Electrónicos Digitales en la titulación de Ingeniería Técnica Telemática, y por extensión, a otras asignaturas afines de otras ingenierías tales como Estructura de Computadores.

## 1. Motivación

Es una problemática muy común en la enseñanza de sistemas digitales, ya sea de electrónica digital o de sistemas basados en microprocesador, la de ofrecer al alumno cuantas herramientas y recursos hardware estén a disposición del profesor para el correcto seguimiento y comprensión de la materia en cuestión. Es evidente que la mejor forma de enseñar, y de que los alumnos comprendan los sistemas digitales, es trabajando con los dispositivos digitales reales que se enseñan en el aula. De este modo, es muy común encontrar en laboratorios de asignaturas de esta índole (Electrónica Digital, Estructura de Computadores, Sistemas Electrónicos Digitales, etc.) todo el equipamiento necesario, como por ejemplo placas de desarrollo, entrenadores, fuentes de alimentación, osciloscopios, etc. Otra opción, es recurrir a la simulación, con las ventajas y restricciones que ello conlleva. Como ventajas destacaremos que la simulación es una herramienta muy potente, puesto que permite evaluar un sistema previo a su montaje y estimar su comportamiento. Así mismo, otra ventaja que

reporta la simulación es su coste, ya que resulta, sin lugar a dudas, menos costosa que equipar un laboratorio con material electrónico. No obstante, la principal desventaja de la simulación es la de que no deja de ser un complemento. Si el sistema simulado no llega a probarse físicamente, el objetivo que se persigue queda, en términos coloquiales, un poco "cojo". La situación ideal es aquella que incorpora parte de simulación y parte práctica real.

El hecho de que al laboratorio se le haya dado un enfoque puramente práctico, aún cuando es muy didáctico, puede suponer un gran impedimento para el alumnado en el desarrollo de las prácticas. Para "investigar" sobre los conceptos explicados en clase o la puesta a punto de las prácticas es imprescindible asistir al laboratorio, puesto que los alumnos no tienen los recursos económicos para adquirir el equipamiento necesario.

En el caso de que el laboratorio sea exclusivamente de simulación, el problema anterior queda subsanado siempre y cuando el simulador en cuestión no precise de una licencia que impida distribuir libremente el software entre el alumnado. En cualquiera de los casos, el alumno nunca llegaría a trabajar con los sistemas reales, que es, en definitiva, con lo que tendrá que enfrentarse en su vida laboral.

Como ya se ha resaltado, el equilibrio idóneo se obtiene de la combinación de simulación y práctica. Sin embargo hay que buscar que haya una correspondencia directa entre lo que se simula y lo que se prueba. En el campo de la enseñanza de electrónica digital y de sistemas basados en microprocesador existe una amplia gama de simuladores de muy diversa índole, en cuanto a sus capacidades y coste. Lo más frecuente es que ciertos simuladores destaquen por unos aspectos

pero que carezcan de otros, es decir, que es muy difícil encontrar un simulador que reúna todas las condiciones que se precisan. En el peor de los casos, no llega siquiera a encontrarse.

Para el caso que nos ocupa, relativo a la enseñanza de sistemas digitales basados en microprocesador, lo ideal sería disponer de un entorno de simulación lo más transparente posible para el alumno. Un entorno en el que trabajase con instrucciones reales, ensambladores reales y depuradores reales y que además que la simulación no tuviese límites. Una de las ventajas que tiene la arquitectura i80x86 frente a otras, es que no es necesario emplear emuladores o entrenadores; cualquier PC es suficiente. Prácticamente todos los alumnos disponen de un PC en casa, con lo que pueden trabajar desde ella. Sin embargo no se puede decir lo mismo cuando las prácticas dejan de ser de mera programación en ensamblador y se pasa a trabajar con dispositivos de entrada/salida (E/S).

Con este ánimo, estamos trabajando en el diseño de simuladores de sistemas periféricos que presentamos a continuación.

## 2. Objetivos

Tomando como base la asignatura Sistemas Electrónicos Digitales, los requisitos que deben cumplir los simuladores a desarrollar son los siguientes:

1. *Transparente al alumno.* El alumno debe tener la sensación de que está trabajando en todo momento con el dispositivo real y debe tener así mismo la certeza de que cuando lo pruebe sobre estos dispositivos obtendrá los mismos resultados. En el ensamblado, enlazado y posterior depuración de su programa debe emplear las mismas herramientas que emplearía para la aplicación real. Nuestros alumnos emplean el programa ensamblador y depurador MacroAssembler [2] y Codeview de Microsoft [3] respectivamente.
2. *Lo más reales posibles.* Los simuladores deben ajustarse lo más posible a los dispositivos que simulan. No deben carecer de ninguna de sus características.
3. *Didácticos.* Que a su vez los simuladores aporten información al alumno que trabaja

con él y que le oriente en caso de que se equivoque en su manejo.

4. *Facilidad de trasladarlo a la realidad.* Que una vez realizada la simulación, el alumno, con unos cambios mínimos, pueda trasladar el código desarrollado al sistema real. Es decir, que no se vean obligados a re-escribir el código.
5. *Gráficos.* Debe ser un simulador gráfico de colores donde se distingan claramente las acciones tomadas por los alumnos en su programa.
6. *Reducido tamaño.* Lo ideal es que no resulten simuladores de gran tamaño. Se persigue que todos los ficheros necesarios para la simulación quepan en un disquete.
7. *Libre distribución.* Los simuladores deben ser gratuitos y abiertos. Al alumno no sólo se le facilitará el ejecutable, sino también su código fuente y una descripción del funcionamiento del mismo.
8. *Ampliables.* ¿Por qué restringir el simulador a una asignatura concreta en la que se estudian unos dispositivos concretos?. Tanto para el profesor como para el alumno es muy interesante que los simuladores sean ampliables.
9. *Multiplataforma.* Se persigue desarrollar simuladores que sean multiplataforma o que, en su defecto, la conversión del simulador de una plataforma a otra sea lo más sencilla posible.

## 3. Desarrollo de los simuladores

Buscando el cumplimiento de los requisitos expuestos en el apartado anterior, se ha pensado en el desarrollo de simuladores modulares, es decir, para cada dispositivo de E/S a simular se desarrollará un módulo diferente. De este modo, la capacidad de ampliación está asegurada. Estos módulos de simulación se programarán en ensamblador, lenguaje con el que trabajan los alumnos en las asignaturas y que conocen perfectamente. En estos programas se cargará por un lado la pantalla gráfica de simulación (VGA 640x480) y por otro lado se definirán los procedimientos necesarios. El alumno desde su programa llamará a esos procedimientos externos definidos en el módulo de simulación (figura 1).

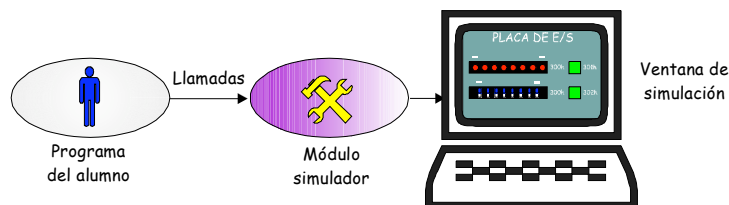


Figura 1. Diagrama de funcionamiento

La arquitectura i80x86, a diferencia de otros microprocesadores, tiene los espacios de direcciones de memoria y entrada/salida separados. Para procesos de E/S existen únicamente dos instrucciones: IN para entrada de puerto y OUT para salida a puerto. Para conseguir la máxima transparencia de cara al alumno, se ha pensado en definir dos instrucciones, llamémoslas virtuales, de entrada/salida que se asemejen lo más posible a las instrucciones reales. Estas instrucciones virtuales son en definitiva dos macros con las que el alumno interactuará con el simulador en cuestión.

Desde el punto de vista del alumno, lo único que cambia respecto a trabajar con un sistema real son las instrucciones de entrada/salida que empleen. Para el trabajo en simulación utilizarán las instrucciones virtuales, mientras que, cuando deseen trasladar su programa a una aplicación real, lo único que deberán hacer es sustituir esas instrucciones virtuales por las reales (operación de buscar y reemplazar en cualquier editor de texto).

Las dos instrucciones virtuales se han llamado INM y OUTM y su formato se corresponde exactamente con el de las originales. En la figura 2, puede apreciarse claramente lo sencillo que le resulta al alumno pasar del entorno de simulación al real.

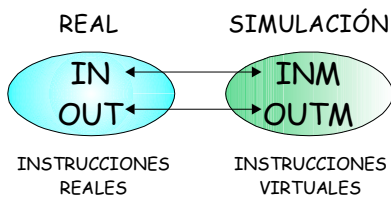


Figura 2. Simulación vs. realidad

Para operar con el simulador, el alumno sólo deberá añadir una pequeña cabecera en su

programa y una vez ensamblado, enlazarlo con el fichero objeto del simulador.

Todos los simuladores desarrollados tienen asignadas unas direcciones de entrada/salida "virtuales" predeterminadas. Con fines didácticos, se ha dotado a los simuladores de capacidad de depuración. De este modo, en caso de que el usuario realice una acción no permitida, bien a propósito o bien por error (como por ejemplo acceder a una dirección errónea), se le informará de ello en pantalla, indicando la causa del error.

Por último, sobre la pantalla del simulador en cuestión se ha incluido el ratón, mediante el cual el alumno podrá interactuar sobre el simulador (cambio de estado de microinterruptores, pulsación de botones, etc.)

#### 4. Simuladores

Hasta el momento, se han desarrollado tres simuladores que se ajustan a los temas tratados en Sistemas Electrónicos Digitales:

- Placa de Entrada/Salida. La ventana de simulación de la placa de E/S se muestra en la figura 3.

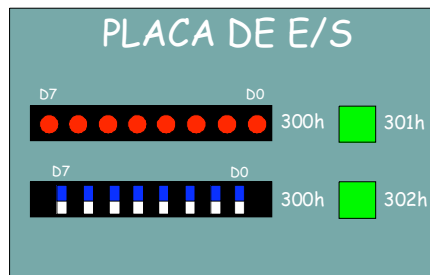


Figura 3. Ventana de la placa de E/S

En ella pueden distinguirse los elementos más característicos como son los micro-

interruptores y los LEDs para simular la entrada y la salida respectivamente.

Tal y como se ha resaltado previamente, las direcciones virtuales de la placa de E/S son fijas, aunque pueden fácilmente cambiarse, reensamblando el programa del simulador. Para el módulo desarrollado se han escogido las direcciones virtuales comprendidas entre la 300h y la 302h. Estas direcciones coinciden con las de la placa de E/S real que existe y que manejan los alumnos en las prácticas de laboratorio.

- Interfaz Paralelo Programable PPI 8255A. La ventana principal de simulación del dispositivo PPI 8255A [1] se muestra en la figura 4.

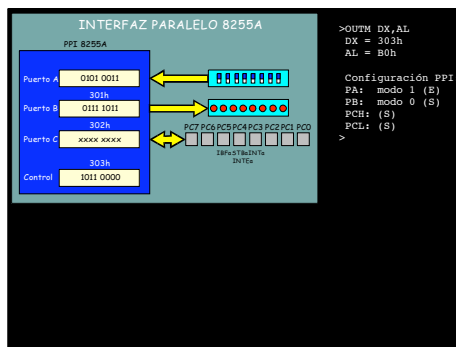


Figura 4. Ventana del simulador de la PPI

Pueden distinguirse claramente los puertos de la PPI, los registros internos y las líneas de control. Se trata de una ventana dinámica. Dependiendo de la configuración de la PPI, las líneas de control se reconfiguran adecuadamente. Así mismo en la parte izquierda de la ventana, el alumno dispone de un histórico de acciones. Si el alumno comete un error, se le informa de ello.

- Interfaz Serie USART 8251A. Como complemento al tema 6 de Sistemas Electrónicos Digitales, se ha desarrollado un simulador de la interfaz serie USART 8251A [1]. La ventana principal de simulación se muestra en la figura 5. Al igual que en el caso de la PPI, se muestran todos sus registros internos y las señales de interés.

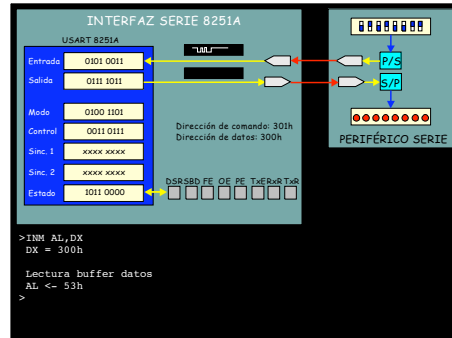


Figura 5. Ventana del simulador de la USART

## 5. Conclusión

Se han presentado tres simuladores de dispositivos periféricos para la familia i80x86. La idea del desarrollo de estos simuladores ha surgido de la experiencia docente y de las sugerencias de los alumnos de Sistemas Electrónicos Digitales. Pensamos que estos simuladores ayudarán a dichos alumnos en el próximo curso académico a comprender mejor la asignatura y servirán como complemento a su laboratorio. Se han desarrollado persiguiendo la máxima versatilidad de los mismos, cumpliendo los requisitos expuestos en el apartado dos. Nuestro propósito es continuar desarrollando nuevos simuladores (PIC 8259, DMA 8237, timer 8254, etc.) que contribuyan a la formación de los alumnos en los sistemas microprocesador. La filosofía en su desarrollo, permite la posibilidad de crear nuevos y potentes simuladores que incluyan una combinación de interfaces programables y periféricos (DACs, ADCs, etc.), siempre persiguiendo simular fielmente la entrada/salida.

## Referencias

- [1] Intel. *Microprocessor and Peripheral Handbook*, Volumen I y II. 1989.
- [2] Microsoft. *Macro Assembler. Programmer's Guide*. 1987
- [3] Microsoft. *Codeview and Utilites*. 1987