

Generador-corrector de problemas de la Unidad de Control

Antonio J. de Vicente¹, Manuel Prieto², Rafael Fernández

Dpto. de Automática

Universidad de Alcalá

Campus Universitario. 28871 Alcalá de Henares (Madrid)

avicente@aut.uah.es, mpm@aut.uah.es

Resumen

Este artículo describe un recurso docente orientado a la impartición del tema de la Unidad de Control de las asignaturas de Arquitectura de Computadores de I. T. Informática de Gestión y la de Arquitectura de Computadores I de I. T. Informática de Sistemas.

El recurso permite generar diferentes computadores elementales, parametrizados por el usuario, y mostrar como serían las fases de ejecución de una instrucción máquina, para distintas instrucciones con diversos modos de direccionamiento. El usuario puede emplear el recurso como un tutorial, o como un generador de problemas para que resuelva, siendo corregidos automáticamente por la aplicación.

1. Introducción

En este artículo, presentamos la aplicación PREVIUC (**P**Resentación **V**isual de la Unidad de Control) orientada al aprendizaje de la unidad de control de un computador. Con ella, el estudiante puede resolver diferentes problemas propuestos por la aplicación, o simplemente emplearla como un tutorial.

La estructura del artículo es la siguiente:

- En el apartado 2 se resume el estado del arte de otros recursos docentes similares.
- El apartado 3 expone la motivación que llevó al desarrollo de la aplicación.
- En el apartado 4 se describen los contenidos del tema de la Unidad de Control de la asignatura de Arquitectura de Computadores dentro del plan de estudios de I. T. Informática de Gestión / Sistemas.
- En el apartado 5 se presentan las diferentes funcionalidades de la aplicación y cómo es su funcionamiento
- El apartado 6 recoge las sugerencias docentes para que el recurso sea empleado.

- En el apartado 7 arroja las conclusiones del desarrollo y empleo del recurso.

2. Estado del arte

El recurso docente que se propone no es nuevo, en la práctica existen una serie de recursos docentes que trabajan el tema de la Unidad de Control.

Entre los recursos que existen se encuentran:

- Spim [1] que permite ver como evoluciona el contenido de la memoria y el de los registros pero que no muestra gráficamente el camino de datos durante la ejecución de la instrucción.
- DLX [2] y su extensión para Windows WinDLX, que si permiten ver de manera gráfica la ejecución de instrucciones sobre el camino de datos.
- SimuProc [3], es un simulador de un Procesador Hipotético con el cual se pueden aprender las nociones básicas para empezar a programar en lenguaje ensamblador. Muestra todo el proceso interno de ejecución del programa a través de cada ciclo del procesador.
- Simula3MS [4], es un simulador que muestra de manera gráfica la evolución de todos los componentes ciclo a ciclo (incluso paso a paso en el caso del procesador multiciclo) y tener también la opción de ejecutar conjuntamente todas las instrucciones y observar únicamente el efecto que produce la ejecución completa.
- MSX88 [5], representa de manera gráfica la ejecución de instrucciones en el Intel 8088. Simula un subconjunto del repertorio de instrucciones de dicho procesador.

3. Motivación

Desde hace varios años, para completar la impartición del tema de la Unidad de Control, hemos sugerido a los alumnos el empleo del programa MSX88. No obstante, su funcionalidad

se quedaba corta, principalmente por la interacción con el usuario, en la que el estudiante era un receptor pasivo del comportamiento de un i8088 simplificado.

Se observó, también, que los alumnos tenían problemas a la hora de desarrollar los cronogramas de ejecución de las instrucciones que se les proponían, y además, el número de ejercicios corregidos les resultaban escasos.

Es por ello, por lo que se desarrolló la aplicación Previuc, en la que el alumno podría parametrizar el computador elemental, entre varias opciones, y ejercitarse en el cálculo de los cronogramas de ejecución para diferentes instrucciones con distintos modos de direccionamiento; sabiendo, que la aplicación le indicaría los errores que pudiera cometer.

4. Descripción de las asignaturas

La asignatura de Arquitectura de Computadores, de la titulación de I. T. Informática de Gestión y de la de Arquitectura de Computadores I de I. T. Informática de Sstemas de la Universidad de Alcalá, se recogen en el B.O.E. del 24 de mayo de 1995 [6].

A partir del descriptor de la asignatura, el tema de la unidad de control ocupa 8 horas de clase, y en él se ven los siguientes conceptos:

1. Operaciones elementales.
2. Estructura de un computador elemental.
3. Señales de control.
4. Temporización de las señales de control.
5. Ejecución de instrucciones.
6. Diseño de la unidad de control.
7. Unidad de control cableada.
8. Unidad de control microprogramada.
9. Microprogramación y nanoprogramación.

La aplicación Previuc abarca los puntos 1 a 5.

5. Previuc

La aplicación Previuc presenta de manera visual la ejecución de instrucciones elegidas por el alumno, en un computador elemental, también diseñado por el alumno. Además, permite al estudiante

resolver los distintos ejercicios de manera interactiva, indicándole la aplicación si ha cometido algún error o si es correcto.

5.1. El asistente para la creación de máquinas

La aplicación cuenta con un asistente para la creación de diferentes computadores elementales con el que el usuario puede decidir que elementos quiere y cuales no en el diseño final de su arquitectura.

La aplicación comienza con dicho asistente, tal y como se muestra en la figura 1.

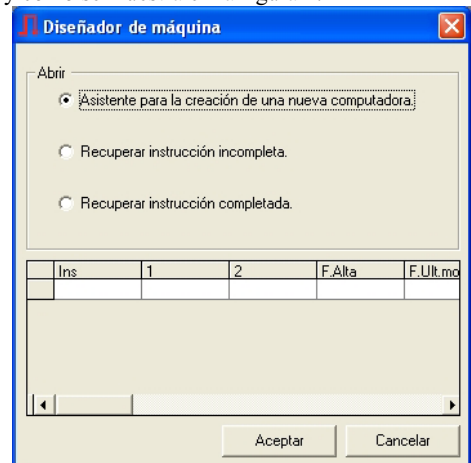


Figura 1. Asistente para la creación del computador.

Cuando se desea crear una máquina nueva sobre la que ejecutar instrucciones, se nos presentan diferentes opciones, recogidas en las figuras siguientes, desde la elección de si el bus es multiplexado para datos y direcciones, o se cuenta con un bus diferente para estos (figura 2)

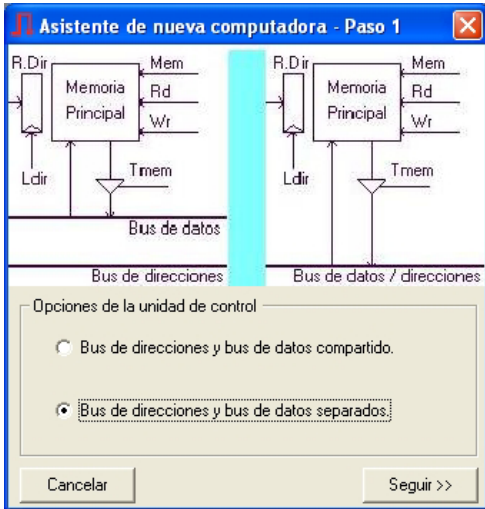


Figura 2. Elección del tipo de bus.

Posteriormente se permite al usuario la elección de si el banco de registros contará con un puerto de entrada y uno de salida o con uno de entrada y dos de salida (figura 3)



Figura 3. Elección del banco de registros.

El siguiente paso es decidir si la unidad de direccionamiento contará con la posibilidad de autoincremento o se deberá pasar por la ALU para actualizar el valor del contador de programa. Dicha elección se muestra en la figura 4.

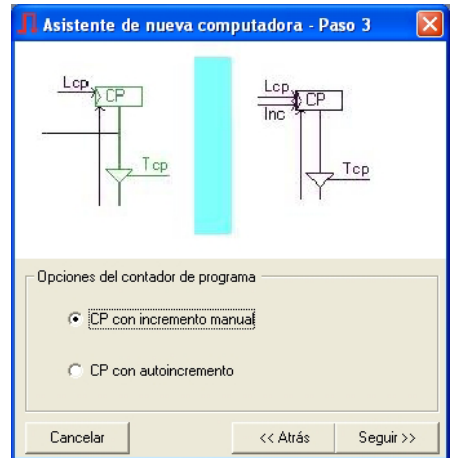


Figura 4. Elección de la unidad de direccionamiento.

Finalmente, se presentan una serie de opciones adicionales para ver si los buses se comunican, o a qué entrada de la ALU se puede conectar el contador de programa (figura 5)

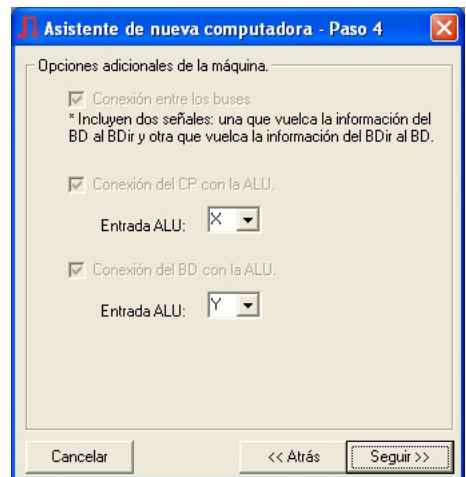


Figura 5. Opciones adicionales.

Si en los 4 pasos anteriores, el usuario ha creado una maquina incoherente, la aplicación generará el computador sin errores que imposibiliten la ejecución de instrucciones

Un ejemplo del computador elemental generado con las opciones anteriores se muestra en la figura siguiente

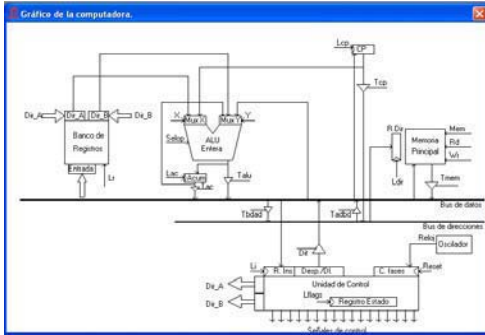


Figura 6. Computador elemental generado.

Una vez generada la máquina elemental, el programa pregunta al usuario por la instrucción que desea ejecutar (figura 7)

Figura 7. Solicitud de la instrucción a ejecutar.

Con la máquina ya terminada y con la instrucción y el modo de direccionamiento elegidos por el usuario, ya permitiría al usuario resolver el cronograma de ejecución (figura 8)

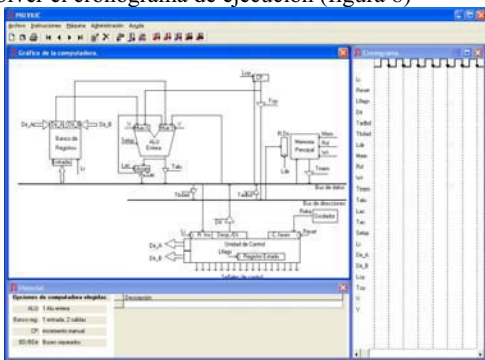


Figura 8. Resolución del cronograma.

La manera de indicar la solución del ejercicio es ir introduciendo las señales y los valores de éstas en cada uno de los periodos de ejecución de la instrucción (figuras 9 y 10)

Figura 9. Introducción de los periodos.

Figura 10. Introducción de las señales de cada periodo de ejecución.

La descripción de lo que hace cada periodo se muestra en la ventana inferior, la del historial, tal y como muestra la figura 11.

Opciones de computadora elegidas.	Descripción
ALU: 1 Alu entera	↪ Vuelve datos desde el CP al bus de direcciones
Banco reg: 1 entrada, 1 salidas	Activa el registro de direcciones con la dirección que hay en el Bdir
CP: incremento manual	
BD/BDir: Bus compartido	

Figura 11. Historial de la ejecución de la instrucción.

El estudiante puede pedir ver la solución entera, o la solución de un determinado periodo y comprobar si lo ha resuelto, o no, correctamente. Los errores se muestran en rojo y la solución de alumno en verde. En la figura 12, los errores se han remarcado con un círculo.

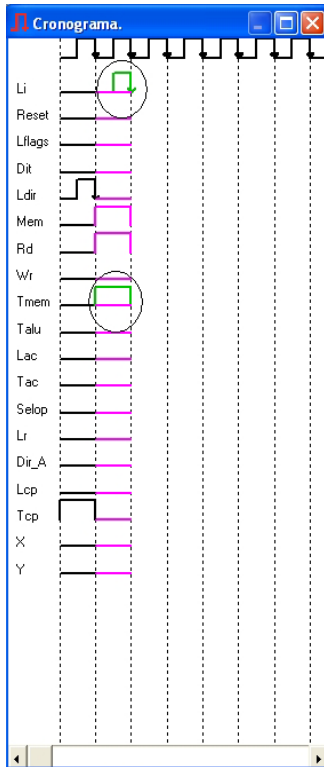


Figura 12. Errores en la solución del alumno.

El alumno puede solicitar en cualquier momento la solución de un periodo o la solución completa y la aplicación se la daría. El cronograma de la solución de la instrucción del ejemplo: **ADD BL, Vector [SI]** se muestra en la figura 13.

La aplicación también permite dejar la resolución del ejercicio para más adelante, con lo que el alumno tiene que guardar el estado actual del mismo. De hecho, guardaría la arquitectura generada y las fases que ha resuelto hasta el momento. De esa manera, al iniciar la aplicación nuevamente, en la pantalla del asistente, el usuario

deberá elegir la opción de “Recuperar instrucción incompleta” (figura 14)

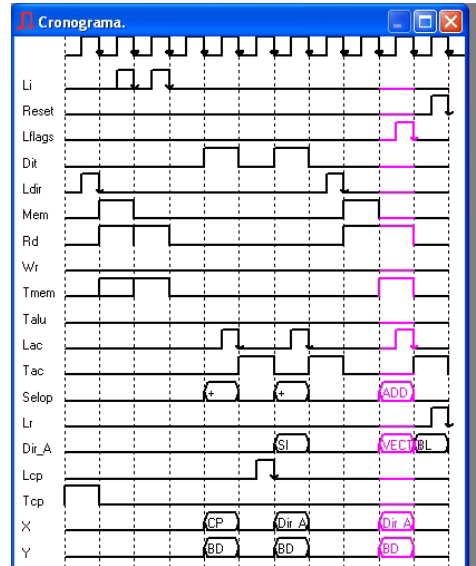


Figura 13. Solución del programa Previuc.

Al recuperar una instrucción incompleta, el programa le mostrará las instrucciones que ha dejado a medias en otras sesiones. Entre ellas el estudiante podrá elegir con cual continuar, simplemente, marcándola de la lista.

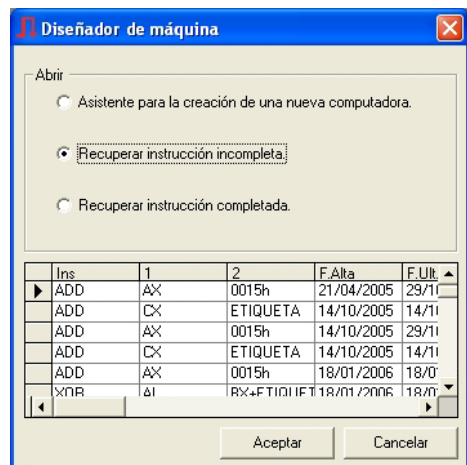


Figura 14. Recuperación de instrucciones incompletas.

Finalmente, para el modo tutorial, la opción de la aplicación es desde el asistente escoger "Recuperar instrucción completada" (figura 15)

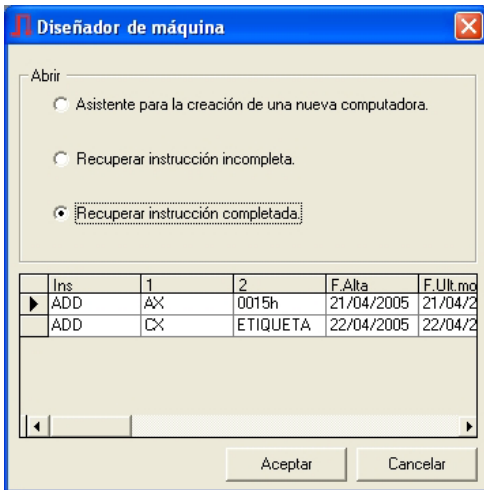


Figura 15. Recuperación de instrucciones completadas.

Una vez que se recupera una instrucción completada, se pueden ver los periodos de los que constó su ejecución, las señales generadas, su cronograma y la explicación del mismo en la ventana del historial.

6. Sugerencias docentes

Es aconsejable que al estudiante se le entregue el programa Previuc junto con una guía metodológica para su aplicación.

En dicha guía se le debe indicar cómo se construye uno de los posibles computadores elementales y las opciones que va a tener disponibles.

También se le debe solicitar la ejecución y entrega impresa de los cronogramas de dos o tres instrucciones con diferentes modos de

direccionamiento (entre los disponibles) para que el profesor evalúe si está empleando el recurso docente de manera correcta o no.

Finalmente, y una vez comprobado que el estudiante no tiene problemas con la aplicación, se le puede dejar sólo resolviendo tantos ejercicios como él desee de manera totalmente autónoma.

7. Conclusiones

Se ha desarrollado una aplicación que permite al estudiante suplir la falta de ejercicios resueltos del tema de la Unidad de Control. Por lo que se espera que en el próximo cuatrimestre la puedan emplear en sus estudios.

No obstante, los autores consideramos que en un futuro se podría desarrollar la misma aplicación, pero para la Web, de manera que sea accesible a todos los estudiantes, incluso, desde sus domicilios y puestos de trabajo.

Referencias

- [1] **Britton, Robert L.** MIPS Assembly Language Programming. Prentice Hall, 2004.
- [2] Boehm, Erich. DLX Distribution Homepage. <http://www.wuwien.ac.at/usr/h93/h9301726/dlx.html>, 1996.
- [3] Simuproc. <http://simuproc.softonic.com/>
- [4] **Concheiro, Raquel ; Loureiro, Marta; Amor, Margarita y González, Patricia.** Simula3MS: simulador pedagógico de un procesador. JENUT'05
- [5] **De Diego Martínez, Rubén.** MSX88: Una Herramienta para la Enseñanza de la Estructura y Funcionamiento de los Ordenadores. URSI'94
- [6] B.O.E. del 24 de mayo de 1995.
- [7] **Patterson, D. A., Patterson J. L.** Estructura y diseño de computadores. Reverté 2000