SiMR: Entorno de simulación de la Máquina Rudimentaria

Fermín Sánchez y Lorenzo Ibarria
Dept. de Arquitectura de Computadores
Universitat Politècnica de Catalunya
Barcelona e-mail: fermin@ac.upc.es

Resumen

En este artículo se describe el funcionamiento de SiMR, un entorno de simulación de un procesador pedagógico: la Máquina Rudimentaria. SiMR permite la edición, compilación y depuración de programas escritos en el lenguaje ensamblador de este procesador. La ejecución de los programas puede simularse con diferentes niveles de detalle. SiMR permite también la confección automática de diversos tipos de informes, lo que facilita la labor del alumno y la posterior evaluación, por parte del profesor, del trabajo realizado.

1. Introducción

La Máquina Rudimentaria (MR) [1,2] es un computador pedagógico. Fue diseñado en la década de los 90 en el Dept. de Arquitectura de Computadores de la Universitat Politècnica de Catalunya para facilitar a los alumnos de primer curso de ingeniería informática el aprendizaje de conceptos básicos sobre arquitectura y estructura de computadores. Es un computador RISC cuyo procesador tiene una arquitectura von Neumann [3] de carga-almacenamiento. La memoria tiene 256 palabras de 16 bits direccionables a nivel de palabra. Las instrucciones son de tamaño fijo de 16 bits y los datos son números enteros codificados en complemento a 2 en 16 bits. La Unidad de proceso (UP) tiene un banco de registros (BR) de 8 registros de 16 bits, un registro contador de programa (PC) de 8 bits y un registro de instrucciones (IR) de 16 bits. El registro de estado tenía originalmente [1,2] los bits de condición N y Z, pero SiMR incorpora también el bit de desbordamiento para enteros V. La Unidad de Control (UC) es muy sencilla y en ella se reflejan perfectamente las fases de ejecución de una instrucción. Hay una versión optimizada (pero semánticamente más compleja) que disminuye el tiempo de ejecución de los programas.

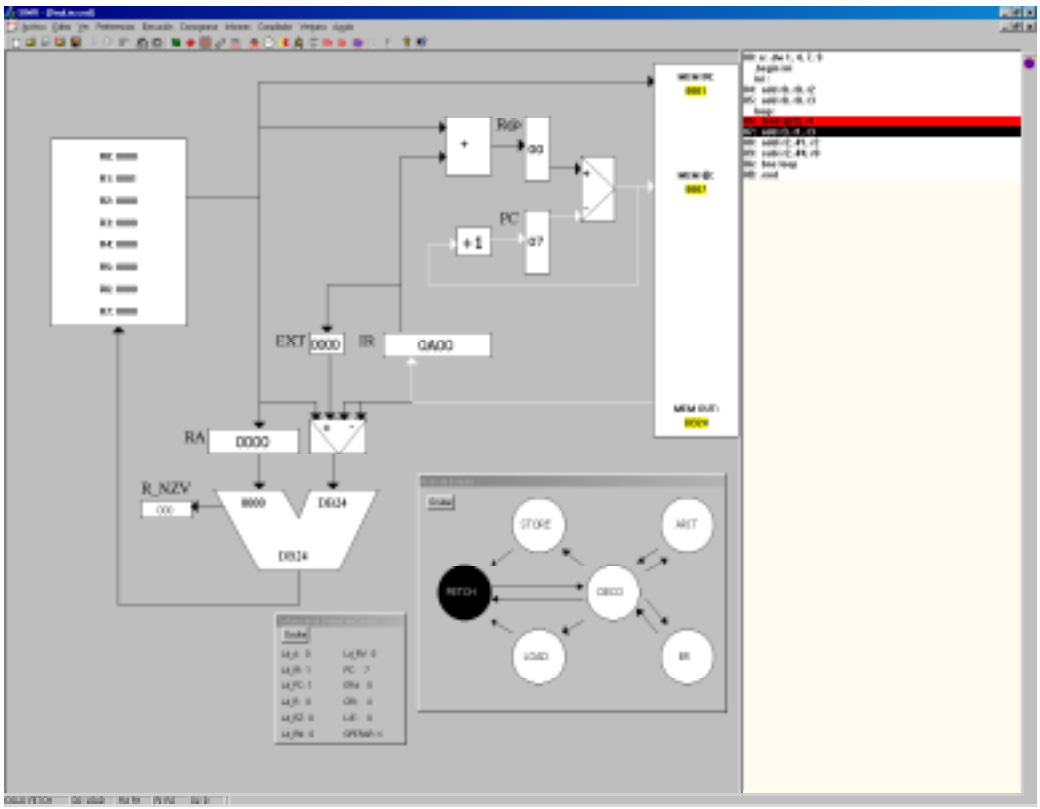
La MR tiene tres tipos de instrucciones:

- Cargas y almacenamientos: una instrucción *load* y otra *store* usan el modo relativo para acceder a la memoria con un desplazamiento de 8 bits (se usan únicamente los 8 bits más bajos del registro índice para el cálculo de la dirección efectiva del operando)
- Aritmético-lógicas: dos instrucciones de suma y dos de resta (una reg-reg y otra reg-inmediato de cada tipo, con un inmediato de 5 bits), una instrucción AND reg-reg y una instrucción de desplazamiento aritmético a la derecha de un registro del BR.
- Saltos: una instrucción de salto incondicional y seis de salto condicional para enteros, que tienen en cuenta los tres bits de condición. Todas utilizan el modo absoluto.

En las siguientes secciones se presenta SiMR, un entorno de trabajo que permite la escritura, compilación, depuración y simulación de programas escritos en lenguaje ensamblador y máquina de la MR. La sección 2 describe las innovaciones principales de SiMR sobre las anteriores versiones de la herramienta. La sección 3 presenta de forma más detallada algunas de estas innovaciones. Finalmente, las conclusiones se presentan en la sección 4.

2. Innovaciones sobre versiones anteriores

SiMR es un entorno construido con la experiencia adquirida en el diseño y utilización de anteriores simuladores de la MR. En 1997 apareció la primera versión, numerada como 1.0. Funcionaba sobre DOS y la pantalla principal estaba dividida en cuatro partes fijas que presentaban la UP (que se podía ampliar para ver las señales de control), la UC, el código del programa que se estaba ejecutando y una última que permitía al usuario gestionar la aplicación. Se *simulaban* programas previamente traducidos por un compilador (no integrado) que se ejecutaba en dos pasos:



1. Precompilador: evaluaba el código de las macroinstrucciones y lo traducía a lenguaje ensamblador de la MR.
2. Postcompilador: traducía el programa a lenguaje máquina.

Un año después, en 1998, se dispuso de la segunda versión, numerada como 2.0. Fue diseñada para correr sobre Windows 3.11 y tenía numerosas ventajas sobre la 1.0. Además de la facilidad de manejo que Windows ofrece frente a DOS (gestión de eventos mediante botones, en lugar de órdenes escritas), incluía la posibilidad de realizar diagramas de tiempo de las señales escogidas y de visualizar, en tiempo de ejecución, el código interno de las macros o simplemente su nombre. La pantalla principal estaba dividida en tres zonas estáticas: una para la UP (que se podía ampliar), otra para la UC y una tercera para el código del programa. El compilador seguía siendo una aplicación separada (el mismo que para la versión 1.0 pero con algunos *bugs* corregidos).

Durante los dos últimos dos años se ha trabajado en la versión 3.0, diseñada para ser ejecutada sobre Windows 95 y posteriores. Esta herramienta, SiMR, es más que un simple simulador: es un entorno completo de trabajo. A diferencia de las dos versiones anteriores, SiMR 3.0 incorpora la posibilidad de utilizar una UC más sencilla que puede usarse desde el comienzo del estudio de la MR. La pantalla principal tiene dos ventanas de tamaño variable, una para la UP y otra para el código (ver figura). La UC aparece como un pequeño cuadro de diálogo aparte.

La interficie con el usuario es muy similar a la de la mayoría de programas que corren sobre Windows, con *Tooltips* en los botones, resalte de las opciones activas y una ayuda moderna y completa, de forma que al alumno puede usar la herramienta de una forma rápida e intuitiva.

El compilador ha sido integrado en SiMR 3.0 junto con un sencillo editor, lo que permite editar, depurar, traducir y simular el programa dentro del

mismo entorno de trabajo. Los dos pasos del compilador son transparentes al usuario y se han corregido *bugs* detectados en versiones previas.

SiMR permite alterar de forma manual el valor de cualquier posición de memoria o registro de la UP, cualidad especialmente útil para usuarios noveles. Esta característica permite al alumno definir un estado inicial de la máquina y ver cuál es el resultado de ejecutar una determinada instrucción. La información de la memoria y el banco de registros se presenta en varios formatos (complemento a dos, binario, decimal, hexadecimal y traducción a lenguaje ensamblador en el caso de la memoria), lo que facilita al alumno la interpretación de los datos.

Es posible deshacer las acciones realizadas en el último ciclo o la ejecución completa de la última instrucción. Esta opción es adecuada para ensayar repetidamente aspectos de la ejecución que no se hayan comprendido correctamente.

SiMR 3.0 permite realizar de forma automática varios tipos de informes que facilitan el trabajo tanto al alumno como al profesor.

3. Descripción de SiMR

En esta sección se describen, brevemente, las características de SiMR 3.0. Como puede observarse en la figura, en la parte superior de la pantalla principal existe una barra de menús y bajo ella se encuentra una barra de herramientas. Los botones permiten acceso rápido a algunas de las opciones (las más frecuentes) de los menús.

Barra de menús

La barra de menús tiene las siguientes opciones:

- *Archivo*: Dispone de las opciones *nuevo, abrir*, *guardar*, *guardar como*, *cerrar* y *salir*. Los archivos son programas escritos en lenguaje ensamblador (*.asm*) o bien ya compilados (*.cod*). Presenta una lista con acceso rápido a los últimos ficheros abiertos.
- *Editar*: Permite *deshacer*, *cortar*, *copiar* y *pegar*. Está activo cuando se está editando un programa en lenguaje ensamblador.
- *Ver*: permite visualizar/ocultar la barra de herramientas (con botones para el acceso rápido a los menús más frecuentes) y la barra de estado, que se encuentra en la parte inferior de la pantalla (ver figura) y presenta información sobre el estado actual de la UC y sobre la instrucción en curso (código de operación y localización de los operandos).
- *Preferencias*: permite variar la frecuencia del reloj y el retardo de los componentes de la UP; guardar, recuperar, ver y alterar el estado de la MR (contenido de la memoria y de los registros de la UP); la configuración (retardo de los distintos elementos) y los colores de la interficie; definir un fichero de macros por defecto, actualizar la identificación del grupo de prácticas y escoger entre las dos UC.
- *Ejecución*: permite seleccionar si la simulación del programa será ciclo a ciclo, instrucción a instrucción o hasta encontrar un punto de ruptura (o se acabe el programa). También puede detener la simulación, insertar puntos de ruptura y poner el simulador en condiciones iniciales.
- *Cronograma*: permite *definir* las señales que se visualizarán en el cronograma (diagrama de tiempos), *ver* un cronograma de la simulación *que se está realizando* (o ya se ha realizado), *guardar* este cronograma, *ver* un cronograma *previamente guardado* o *imprimir* un cronograma.
- *Informes*: se pueden *escribir*, *imprimir* o *visualizar* los diversos tipos de informes que SiMR prepara de forma transparente al usuario. Los informes contienen información sobre el proceso de *compilación* (programa ensamblador y máquina, tabla de símbolos y errores de compilación), *actividad* (se guarda una traza de los pasos seguidos por el alumno, de forma que el profesor puede seguir paso a paso la realización de cada práctica), *ejecución* (guarda solo los pasos relacionados con la ejecución de un programa) y *resultados* (estado de la máquina antes y después de la simulación).
- *Compilador*: permite *compilar*, *compilar y ejecutar* (si no se han producido errores de compilación) y *ver la Tabla de Símbolos*.
- *Ventana*: Sirve para organizar las ventanas abiertas. Permite situarlas en *cascada, mosaico* o *mosaico vertical*. La parte inferior del menú permite un acceso rápido a las ventanas abiertas en el momento actual.
- *Ayuda*: Accede a una ayuda *on-line* muy potente y completa y suministra información técnica sobre el simulador.

Barra de herramientas

La barra de herramientas permite:

- *Crear*, *abrir* o *guardar* archivo de programa
- *Cortar*, *copiar* y *pegar* trozos de texto en un archivo de programa
- *Deshacer* la ejecución de la última instrucción o el último ciclo de ejecución
- *Ver/ocultar* las señales de control en la UP
- Acceder al cuadro de diálogo que permite *ver y editar el contenido de la memoria, el BR y los distintos registros de la UP*
- *Ver un cronograma* de las señales, registros y buses predefinidos por el usuario.
- *Inicializar* el simulador (con el programa actual cargado en memoria)
- *Avanzar la simulación* en un ciclo, una instrucción o hasta que se encuentre el primer punto de ruptura (o el final de programa)
- *Detener* la ejecución del programa en curso
- *Poner o quitar un punto de ruptura*
- *Compilar o ejecutar* (compilando si es necesario) el programa en curso
- Dar *información* sobre el simulador
- *Ayuda* interactiva

Otras características de SiMR

Además de las características citadas en las secciones anteriores, SiMR 3.0 permite *expandir/comprimir* las macros mediante un botón situado estratégicamente en la ventana de código.

La ventana de código tiene, además, ciertas características especiales: posicionándose sobre una línea cualquiera del programa y pulsando el botón derecho del ratón aparece un menú que permite insertar o eliminar un punto de ruptura o un *stop*. Un *stop* es un punto de ruptura temporal, que desaparece una vez el programa se ha detenido en esa instrucción.

El cuadro de diálogo de la UC permite seleccionar entre dos posibles UCs: una con pocos estados, ideal para usuarios avanzados, y otra más "grande" y semánticamente más sencilla en la que se ilustran perfectamente las fases de ejecución de una instrucción, ideal para usuarios noveles.

Haciendo clic sobre los distintos componentes de la UP puede alterarse su valor (si son dispositivos secuenciales) en cualquier momento de la ejecución y también su tiempo de respuesta.

Forma de trabajar con SiMR

La forma de trabajar con el entorno es sencilla e intuitiva. Cuando se ejecuta el programa aparece, en primer lugar, un cuadro de diálogo que solicita información sobre el grupo de alumnos que va a hacer la práctica. Este cuadro puede saltarse si se está simplemente estudiando, pero debe rellenarse si se está haciendo una práctica, ya que la información que se introduzca aparecerá en todos los informes. Éstos no deberán ser editados con posterioridad, ya que SiMR dispone de un sistema de seguridad para detectarlo.

A continuación aparece la pantalla principal (ver figura). Esta pantalla presenta la UP y la UC, pero es preciso editar un programa nuevo o cargar uno ya existente para comenzar simular. Una vez el programa ha sido cargado o compilado, el simulador lo almacena a partir de la posición 0 de memoria (a menos que el programa especifique lo contrario), inicializa el PC con la dirección de la primera instrucción a ejecutar, selecciona el estado de *FETCH* en la UC y deja control al usuario para que ejecute el programa como desee. Todas las operaciones que el usuario realiza quedan reflejadas, de una u otra forma, en los informes de trabajo que SiMR genera de forma automática.

4. Conclusiones

En este artículo se han presentado las características de SiMR 3.0, un entorno de trabajo sobre Windows 95 que integra un compilador y un simulador de la máquina Rudimentaria. SiMR ha sido diseñado para ayudar a un estudiante de Ingeniería en Informática (técnica o superior) a comprender el funcionamiento de un procesador.

Referencias

[1] R.Hermida, A.M.del Corral, E.Pastor y F. Sánchez. *Fundamentos de Computadores.* Ed. Síntesis, 1998

[2] E.Pastor y F.Sánchez. La Máquina Rudimentaria: Un procesador pedagógico. III Jornadas de Enseñanza Universitaria sobre Informática JENUI'97 , Junio 1997, pag. 395-402.

[3] A.W.Burks, H.H.Goldstine y J.von Neumann. *Preliminary discussion of the logical design of an electronic computing instrument.* A.H.Taub ed., collected works of John von Neumann, vol. 5, pag. 34-79, The Macmillan Company, 1963.