

APLICACIÓN DEL CONJUNTO COMPUTADORA SIMULADA-SISTEMA MICROENSAMBLADOR A LA ENSEÑANZA

Benavides, J. Ignacio; Saez, Edmundo; Quiles, Fco. Javier;
Montijano, Miguel A. y Ortiz, Manuel A.

Grupo Arquitecturas Avanzadas de Computadores, Universidad de Córdoba
E.U. Politécnica, Avda. Menéndez Pidal, s/n, 14004 – Córdoba

Tfno: 957 218375(6), Fax: 957 218316, e-mail: ellbebej@uco.es

RESUMEN

En este trabajo se presenta una herramienta que, junto con el software de simulación de una computadora didáctica diseñado anteriormente por este grupo de trabajo, posibilita profundizar en la enseñanza de asignaturas relacionadas con la arquitectura de computadores. Se trata de un sistema microensamblador que permitirá al alumno pasar a tomar parte activa, no sólo en la creación de programas a partir de instrucciones, lo cual ya ofrecía el software de simulación de la computadora didáctica, sino también en la creación de instrucciones a partir de microinstrucciones, que podrán ser usadas posteriormente para implementar cualquier programa que desee usarlas.

1. INTRODUCCIÓN

En este trabajo se presenta una herramienta con fines docentes para su aplicación en las enseñanzas relacionadas con el diseño de procesadores. Se trata de la aplicación de un Simulador de una computadora microprogramada tipo CISC junto con un microensamblador que permite analizar o modificar Instrucciones de esa computadora CISC, o incluso crear nuevas.

A la hora de la implementación del microensamblador se han tenido presente los siguientes objetivos: contenido didáctico; nivel acorde con el de los alumnos con el que se quiere aplicar; facilidad de uso; bajo coste y uso de hardware estándar.

Un buen conocimiento tanto de la estructura de la Computadora CISC, así como de la técnica de Microprogramación serán imprescindibles para que el alumno pueda llegar a obtener un buen rendimiento de aprendizaje con esta herramienta. En los apartados siguientes, expondremos la Estructura de la Computadora, el Simulador y el Microensamblador. Un apartado analizará la Asignatura sobre la que se está usando, y acabaremos con uno dedicado a las conclusiones sobre esta experiencia.

2. ESTRUCTURA DE LA COMPUTADORA CISC

En la figura 1 se muestra dicha estructura, donde podemos ver las unidades funcionales que la componen: Unidad de Cálculo (UCA), Unidad de Gestión de Memoria (UGM), Unidad de Control (UCO), y Unidad de E/S (UES). Dichas Unidades se encuentran intercomunicadas mediante los buses: de Direcciones (16 bits), de Datos (8 bits), y el Bus Interno (8 bits). La UES no ha sido incluida en el Simulador.

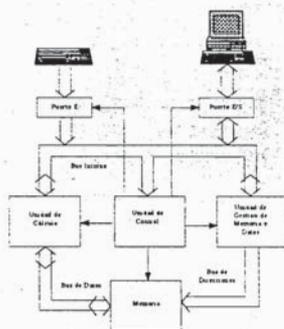


Figura 1. Diagrama de bloques de la computadora

2.1. Unidad de Cálculo

Básicamente está formada por un Unidad Aritmético-Lógica capaz de implementar hasta 32 operaciones diferentes y un conjunto de registros. En la figura 2 se muestra dicha Unidad tal y como aparece en el Simulador.

Entre los registros más importantes, todos de 8 bits, señalaremos:

- Registros Acumuladores: Se subdivide en tres registros: A, B y E, que pueden ser fuentes y destino en las operaciones de la ALU.
- Registro de Estado: Formado por los bits de acarreo (C), negativos (N1, N2, N3), de cero (Z) y de overflow (V).
- Registro de Índice: Se usa principalmente para direccionamiento indexado.
- Registro de salida de la ALU.
- Registro Temporal de propósito general.

Estos registros se encuentran unidos a diferentes buses internos de la Unidad de Cálculo como son: el bus de entrada (A) y de salida (S) de la ALU, y algunos de ellos al Bus Interno de la Unidad Central de Procesamiento (B).

2.2. Unidad de Gestión de Memoria

Tal y como se muestra en la figura 3, está formada por el bloque de memoria de 64 Kbs. De lectura/escritura y acceso aleatorio y un conjunto de Registros de 16 bits. Estos son: el Contador de Programa (PC), el Puntero de Pila (SP) y los registros de dirección RD1 y RD2. Todos ellos, salvo el RD2, pueden acceder al Bus de direcciones de memoria.

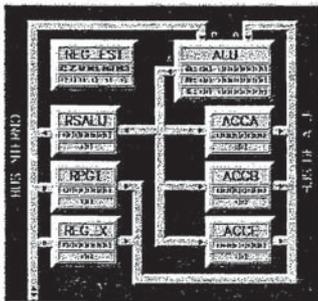


Figura 2. Unidad de Cálculo

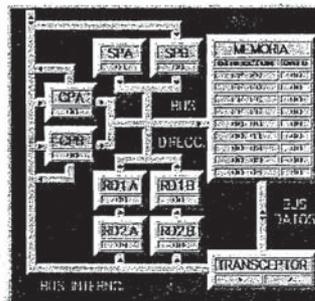


Figura 3. Unidad de Gestión de Memoria

2.3. Unidad de Control

La Unidad de Control simulada es microprogramada, aunque en la primera versión que se diseñó de la Computadora CISC se implementó físicamente, siendo simuladas por ordenador las restantes. En la figura 4 se muestra un diagrama de bloques de esta unidad donde se pueden observar los diferentes módulos, entre ellos la ROM de Control.

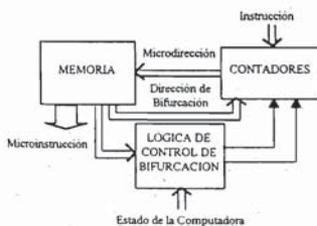


Figura 4. Unidad de Control

EPROM 1							
D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
IS	LRD2B	LRD2A	LRD1	INCD1	UGM0	UGM1	UGM2
LACCE	LRSALU	ALU0	ALU1	ALU2	ALU3	ALU4	ALU5
EPROM 2							
D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
R/W	CBD0	CBD1	CB0	CB1	CB2	CB3	CB4
Y0	Y1	X0	X1	ACC0	ACC1	ACC2	ACC3
EPROM 3							
D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
DV/DA	CB10	CB11	CB12	CBA0	CBA1	CBA2	HT
LRS2	LRE2	LRE1	AV	N0	N1	C0	C1

MD: MICRODIRECCION
 CB: CONTROL DE BIFURCACION
 CBD: CONTROL DE BUS DE DIRECCIONES
 CDA: CONTROL DE BUS DE ALU
 CBI: CONTROL DE BUS INTERNO

Figura 5. Contenido de la CROM

El primer aspecto a destacar es que, para disminuir la longitud de la micropalabra, la ejecución de una microinstrucción se realiza en dos ciclos de reloj. En el primer ciclo realiza las operaciones con la UGM y en el segundo ciclo las operaciones en la UCA. Además, se ha intentado minimizar la longitud de la palabra codificando las señales convenientemente.

En la figura 5 se muestra el significado de la micropalabra, teniendo en cuenta que es de 24 bits, divididos en tres micropalabras de 8 bits cada una. Cada grupo de 8 bits tiene su significado en los dos ciclos de una microinstrucción. Además, los primeros 8 bits tienen un significado de "dirección de salto" si $IS = 1$.

En la pantalla del Simulador se muestra una ventana donde aparecen las señales de control agrupadas de forma que se activen cada vez que se usan en alguno de los dos ciclos de cada microinstrucción, para poder seguir las secuencias de cada instrucción.

3. ENTORNO SIMULADOR

El programa Simulador de esta computadora presenta una pantalla donde aparece la estructura comentada anteriormente (UGM, UCA y UCO), y un menú de opciones entre los que cabe destacar:

- *Memoria*: permite la introducción de instrucciones y/o datos en Memoria, desde el teclado o archivo y almacenar datos y/o programas en archivos.
- *Ejecución*: permite seleccionar cualquiera de los tres modos de ejecución que tiene el Simulador (programa completo, instrucción a instrucción, y ciclo a ciclo).
- *Ayuda*: da información sobre las posibles ayudas del Programa.
- *Salir*: permite abandonar la ejecución del Programa.

Además posee un conjunto de botones con las opciones de: reset, ejecución pulso a pulso, visualización de distintos bloques (UGM, UCA), etc.. El manejo del Simulador se puede resumir en los siguientes pasos:

- 1.- Por medio de la opción Datos, introducimos un programa en memoria.
- 2.- Por medio de la opción Ejecución, elegimos el modo de los tres posibles, que en nuestro caso será el modo Ciclo a Ciclo.
- 3.- Pulsando el botón de ciclo se irán ejecutando las instrucciones ciclo a ciclo, activándose las señales apropiadas en la ventana de la UCO.

4. ENTORNO MICROENSAMBLADOR

Se ha desarrollado un sistema microensamblador basado en un interface gráfico, donde todas las señales de la CROM estén a la vista, así como una ayuda que describe los distintos valores que pueden tomar y la acción que implican. En la figura 6 se observa la pantalla. En las ventanas centrales, se muestran todas las señales correspondientes a los dos ciclos de una microinstrucción y en ambos laterales diferentes ventanas que con las distintas opciones: seleccionar Instrucción, siguiente microinstrucción, etc.

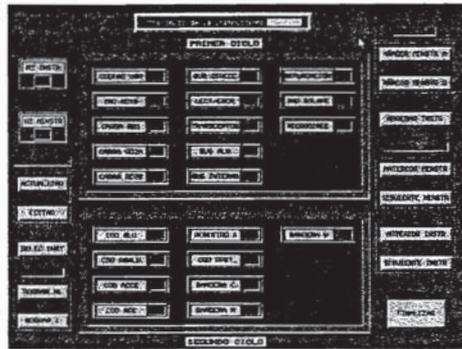


Figura 6. Pantalla del microensamblador

Básicamente las opciones que nos ofrece el Microensamblador son:

- 1.- Análisis de la secuencia de microinstrucciones, viendo los valores que van tomando las señales y el contenido de la CROM, para cada una de las instrucciones definidas.
- 2.- Modificar una instrucción ya existente, introduciendo los cambios apropiados en las señales correspondientes en el ciclo que se desee.
- 3.- Definir nuevas instrucciones, introduciendo los valores para las distintas señales en todos los ciclos.

	Microinstrucción		
	1	2	3
INC RD1B	3	0	3
CARGA RD1	0	0	0
CARGA RD2A	0	0	0
CARGA RD2B	0	0	0
BUS DIRECC	0	1	0
LECT/ESCR	1	1	1
TRANSECTOR	1	0	1
BUS ALU	0	0	0
BUS INTERNO	0	0	0
BIFURCACION	1	1	0
IND SOLAPE	1	1	1
MICRODIREC	0	0	0
COD ALU	26	26	26
COD RSALU	0	0	0
COD ACCE	0	0	0
COD ACC	15	15	15
REGISTRO X	0	0	0
COD RPTG	0	0	0
BANDERA C	0	0	0
BANDERA N	0	2	0
	0	0	0

Tabla I. Microcódigo de la instrucción carga inmediata en el acumulador

	Microinstrucción							
	1	2	3	4	5	6	7	8
CODIGO UGM	3	0	3	0	0	0	0	3
INC RD1B	0	0	0	0	0	0	0	0
CARGA RD1	0	0	0	1	0	0	0	0
CARGA RD2A	0	0	0	1	0	0	0	0
CARGA RD2B	0	1	0	0	0	0	0	0
BUS DIRECC	0	1	0	1	2	0	2	0
LECT/ESCR	1	1	1	1	1	1	0	1
TRANSECTOR	1	0	1	0	0	1	0	1
BUS ALU	0	0	0	0	0	5	0	0
BUS INTERNO	0	0	0	0	0	0	3	0
BIFURCACION	1	1	1	1	1	1	1	0
IND SOLAPE	1	1	1	1	1	1	1	1
MICRODIREC	0	0	0	0	0	0	0	0
COD ALU	26	26	26	26	26	48	26	26
COD RSALU	0	0	0	0	0	1	0	0
COD ACCE	0	0	0	0	0	0	0	0
COD ACC	15	15	15	15	15	15	15	15
REGISTRO X	0	0	0	0	0	0	0	0
COD RPTG	0	0	0	0	0	0	0	0
BANDERA C	0	0	0	0	0	0	0	0
BANDERA N	0	0	0	0	0	3	0	0
BANDERA V	0	0	0	0	0	0	0	0

Tabla II. Microcódigo de la instrucción de complemento a dos

En la tabla I se muestran los valores que presentan las señales, para el ciclo de ejecución, para instrucción "Carga de dato inmediato en el Acumulador A", en cada una de las 3 microinstrucciones que la componen. La primera incrementa el Contador de Programa. La segunda activa la señal de lectura, habilita el Buffer y el dato, que pasa a través del Bus

Interno, y la ALU se carga en A. Y la tercera incrementa de nuevo el Contador de Programa. En la tabla II se muestra la secuencia de microinstrucciones que componen la instrucción nueva definida de cálculo del complemento a dos de una posición de memoria.

5. APLICACION A LA DOCENCIA

En el pasado curso académico, el sistema Microensamblador junto con el Simulador de la Computadora CISC se ha utilizado en la realización de las sesiones prácticas de la asignatura "Diseño de Computadores" de los estudios de I.T. en Electrónica Industrial. Las prácticas han sido de tres tipos:

- 1) Análisis de la secuencia de microinstrucciones de instrucciones definidas.
- 2) Modificaciones a introducir en el contenido de la CROM para cambios en la UCA.
- 3) Diseño de nuevas instrucciones.

6. CONCLUSIONES

El empleo del combinado Microensamblador-Simulador de la computadora CISC, ha posibilitado que los alumnos puedan conocer con profundidad cómo es posible el diseño de Instrucciones de un procesador con la técnica de microprogramación. Sin duda este grupo de alumnos, además de ser reducido, lo que facilita la calidad de la enseñanza, había ya cursado otras asignaturas de la misma área donde habían utilizado simuladores de arquitectura sencilla y habían analizado secuencias de microoperaciones para un conjunto sencillo de instrucciones.

Quizás la principal dificultad que se ha encontrado en la forma concreta de cómo se ejecutan las microinstrucciones y el desarrollar cada una de ellas en dos ciclos. El hecho de que cada alumno haya tenido que analizar diferentes instrucciones de complejidad creciente, ha hecho posible que puedan comprender a partir del análisis correspondiente, la secuencia de microoperaciones de instrucciones como suma, resta, multiplicación y división en punto flotante, que están incluidas en el Simulador y en el Microensamblador.

Así pues, el valor docente de esta herramienta es, a nuestro juicio, alto, dado que aborda el tema de la microprogramación, que es poco tratado por las herramientas software existentes.

7. REFERENCIAS BIBLIOGRÁFICAS

- [1] W. Stallings. "Organización y Arquitectura de Computadoras". Limusa. S.A. de C.V. Grupo Noriega Editores. 1995.
- [2] D.A. Patterson y otro "Organización y diseño de Computadoras". Mc.Graw Hill. 1995.
- [3] S. Gracia, A. Peñas. "Herramienta para el estudio interno de un Computador: Control simulado de ejecución. Sistema Microensamblador". E.U.P. U. Córdoba. 1997.