

## UN SIMULADOR DE UNA MAQUINA COMPUTADORA COMO HERRAMIENTA PARA LA ENSEÑANZA DE LA ARQUITECTURA DE COMPUTADORAS

### **Autores**

GROSSI, María Delia ([mdgrossi@mara.fi.uba.ar](mailto:mdgrossi@mara.fi.uba.ar))

JIMÉNEZ REY, M. Elizabeth ([ejimenez@mara.fi.uba.ar](mailto:ejimenez@mara.fi.uba.ar))

SERVETTO, Arturo Carlos ([aserve@mara.fi.uba.ar](mailto:aserve@mara.fi.uba.ar))

PERICHINSKY, Gregorio ([gperi@mara.fi.uba.ar](mailto:gperi@mara.fi.uba.ar))

Paseo Colón N° 850, 4° Piso, Tel. 4343-0891 Int. 142

*Departamento de Computación, Facultad de Ingeniería, Universidad de Buenos Aires*

### **Resumen**

En este trabajo se presenta una experiencia educativa específica para la Enseñanza de la Arquitectura de Computadoras, basada en el uso de un Programa Simulador del funcionamiento de una máquina computadora genérica que permite a los alumnos resolver problemas adaptándose a las limitaciones del lenguaje de máquina.

Se caracteriza la situación didáctica, se fundamentan los propósitos de la utilización del programa simulador, se describen sus características y se exponen los resultados obtenidos.

### **Palabras clave**

Máquina genérica; simulación; instrucciones de máquina; funcionamiento CPU; manipulación de datos.

### **Introducción**

El objetivo de este trabajo es mostrar cómo se aborda la enseñanza y el aprendizaje de la arquitectura y el funcionamiento de computadoras a través del diseño del módulo “Manipulación de Datos”.

Los autores son docentes de la materia Computación de la Facultad de Ingeniería de la Universidad de Buenos Aires. En la institución esta materia es de formación básica obligatoria para alumnos de todas las carreras de Ingeniería, excepto para aquellos que cursan Ingeniería Electrónica e Ingeniería en Informática. Son sus contenidos mínimos: Arquitectura de Computadoras. Software de Sistema, de Aplicación y de Traducción. Lenguajes de Programación. Algoritmia y Programación Básicas.

### *Conocimiento previo*

En este estadio de aprendizaje los alumnos han adquirido conocimientos de algoritmia y de algunas técnicas de representación y almacenamiento de información dentro de las computadoras.

Los alumnos han incorporado la idea general de que la solución de cualquier problema algorítmico se basa en el proceso de resolución de problemas definido por el matemático Polya que consta de las siguientes fases: 1) Comprender el problema. 2) Idear un plan para resolver el problema (determinar los recursos para resolver el problema y cómo utilizarlos). 3) Llevar a cabo el plan (emplear una estrategia para resolver el problema). 4) Evaluar la solución en cuanto a su exactitud y a su potencial como herramienta para resolver otros problemas.

Los alumnos han asimilado los conceptos básicos relacionados con la Memoria Principal, Almacenamiento Masivo, Codificación de la Información para su almacenamiento, El Sistema Binario, Almacenamiento de Enteros, Almacenamiento de Fracciones.

#### *Conocimiento nuevo*

Se presenta a los alumnos las formas en que las máquinas actuales manipulan los datos bajo la dirección de un algoritmo. Se estudian los siguientes temas: La unidad central de proceso. El concepto de programa almacenado. Ejecución de programas. Otras arquitecturas. Instrucciones de máquina. Comunicación entre el computador y los periféricos.

### **Caracterización de la Situación Didáctica**

#### *Contenido*

Los objetivos del módulo son [1]:

- Que los alumnos conozcan la arquitectura y el funcionamiento de la computadora para que puedan aprovecharla como herramienta de trabajo conociendo sus capacidades y limitaciones.
- Que los alumnos conozcan tecnología y vocabulario informático para interactuar con profesionales de la informática en su vida profesional
- Que los alumnos se entrenen en la resolución de problemas adaptándose a las limitaciones del lenguaje de máquina para desarrollar sus capacidades de abstracción y de resolución de problemas en general.

Los principios conceptuales son:

- Un programa es la representación de un algoritmo para que pueda interpretarlo una computadora.
- Las instrucciones de un programa y los datos que éste procese se almacenan en la Memoria Central.
- Las instrucciones de los programas son interpretadas y llevadas a cabo por la Unidad Central de Proceso de la computadora.
- Un registro es una celda de memoria especial que se encuentra dentro de la Unidad Central de Proceso.
- Un registro sirve para guardar temporariamente los datos que se manipulan.
- Una máquina genérica contiene tres tipos de instrucciones: de transferencia, Aritmético – Lógicas y de Control.
- La comunicación con periféricos o entrada/salida de datos en un programa se realiza mediante instrucciones de transferencia entre CPU y puertos de entrada/salida, que según la arquitectura de la máquina (CISC o RISC) pueden ser específicos (com1, com2, lpt1, lpt2) o corresponderse con celdas de memoria.

Los principios procedimentales son:

- Para desarrollar un programa en lenguaje de máquina se deben seguir las fases de Polya aplicadas a la programación, igual que para desarrollar programas en lenguajes de alto nivel: análisis, refinamientos sucesivos y prueba de escritorio, codificación y depuración, y evaluación.

- El desarrollo de un programa implica descubrir un algoritmo y luego representarlo en forma de programa.
- Para comunicarse con un periférico se requiere sincronizar el programa con un usuario (para entradas) o con un dispositivo de salida (monitor o impresora).

#### *Docente*

El curso está a cargo de un profesor y dos docentes auxiliares. Se compone de sesenta alumnos. La relación docente-alumno es mala.

#### *Alumno*

Los alumnos que cursan la materia pertenecen a distintas carreras de Ingeniería y pueden hacerlo en distintas etapas de avance en sus planes de estudio. La mayoría de los alumnos la cursan al principio de la carrera cuando el desarrollo de la capacidad de abstracción es aún baja.

#### *Condicionantes*

El tiempo de desarrollo del módulo resulta condicionado por la relación entre contenidos y la carga horaria de la materia.

### **Descripción del Simulador de Máquina Genérica**

#### *Arquitectura de la Máquina*

La máquina genérica a simular tiene 16 registros designados de 0 a F (en base hexadecimal). Cada registro tiene un byte de longitud.

La memoria principal consta de 256 celdas de un byte, con direcciones representadas con patrones hexadecimales entre 00 y FF. Los puertos de entrada/salida se consideran en correspondencia con la memoria: FC, FD para entrada de datos, y FE, FF para salida. En ambos puertos, la primera celda es para control y la segunda para datos.

Para poder leer del puerto de entrada su celda de control debe contener el patrón 01 indicando que hay un dato para consumir, y luego de la carga del dato se debe almacenar en ella el valor 00 indicando que se consumió; para poder escribir en el puerto de salida el esquema es inverso: el consumidor es el dispositivo de salida y el programa es el que produce, así que se debe almacenar el dato en la celda de datos del puerto de salida y luego almacenar el valor 01 en la celda de control para indicar que el dispositivo tiene un dato para consumir.

Los valores de punto flotante se almacenan en 8 bits, de los cuales el primero (más significativo o del extremo alto) se emplea para el signo, los tres siguientes para el exponente en exceso de 4, y los cuatro restantes (menos significativos o del extremo bajo) para la mantisa.

Cada instrucción de la máquina tiene 2 bytes de longitud (ocupa dos celdas). Los primeros cuatro bits constituyen el código de operación, los 12 bits restantes constituyen los operandos de la instrucción.

A continuación se describen las instrucciones de máquina en notación hexadecimal. Se utilizan las letras R, S y T para representar un dígito hexadecimal identificador de registro. Las letras X e Y se usan en lugar de dígitos hexadecimales para representar valores o direcciones de memoria[2].

### *Instrucciones*

- 1RXY CARGAR el registro R con el patrón de bits que está en la celda de memoria cuya dirección es XY
- 2RXY CARGAR el registro R con el patrón de bits XY
- 3RXY ALMACENAR el patrón de bits que está en el registro R en la celda de memoria cuya dirección es XY
- 4ORS COPIAR el patrón de bits que está en el registro R al registro S
- 5RST SUMAR los patrones de bits de los registros S y T como si fueran representaciones en complemento a 2 y dejar el resultado en el registro R
- 6RST SUMAR los patrones de bits de los registros S y T como si representaran valores en notación de punto flotante y dejar el resultado de punto flotante en el registro R
- 7RST Disyunción lógica (OR) de los patrones de bits de los registros S y T colocando el resultado en el registro R
- 8RST Conjunción lógica (AND) de los patrones de bits de los registros S y T colocando el resultado en el registro
- 9RST Disyunción lógica exclusiva (XOR) de los patrones de bits de los registros S y T colocando el resultado en el registro R
- AR0X ROTAR el patrón de bits del registro R un bit a derecha X veces
- BRXY SALTAR a la instrucción situada en la celda de memoria cuya dirección es XY si el patrón de bits del registro R es igual al patrón de bits del registro número 0.
- C000 PARAR la ejecución

### *Requerimientos*

Se requiere Windows 95 o superior.

### *Forma de uso[3]*

En un archivo de texto se debe escribir el programa en lenguaje de máquina que se desea ejecutar.

Las instrucciones del programa se deberán escribir siguiendo la siguiente convención:

A partir del primer renglón del archivo de texto y en tantos renglones como instrucciones tenga el programa se deberá escribir para cada instrucción: la dirección de memoria en donde se almacenan los primeros 8 bits de la instrucción en hexadecimal, debiendo ser la primera siempre 00; la instrucción en hexadecimal separada de la dirección por uno o más espacios en blanco; y opcionalmente un comentario en lenguaje natural, separado de la instrucción por uno o más espacios en blanco.

El simulador solicitará el nombre del archivo donde se almacenó el programa.

Si el programa realiza operaciones de entrada de datos, en pantalla aparecerá el mensaje “Entra:” para que a continuación el usuario ingrese un valor de dos dígitos hexadecimales (\$dd).

Si el programa realiza operaciones de Salida, en pantalla aparecerá el mensaje “Sale:” seguido de dos dígitos hexadecimales.

### **Modalidad de Aprendizaje**

Cada grupo de alumnos conformado por tres integrantes deberá desarrollar un trabajo práctico que consiste en la escritura en lenguaje de máquina de tres programas.

*Ejemplificación de programas a desarrollar:*

1. Leer dos números en punto flotante del teclado, calcular el producto del primero por el segundo y escribir el resultado en la pantalla.
2. Leer cuatro patrones del teclado y agruparlos de a dos considerando que conforman, en orden de lectura, las mitades más y menos significativas de un patrón en complemento a dos, hacer la suma aritmética de ambos patrones y escribir el resultado en pantalla. Tener en cuenta que para sumar las mitades menos significativas de ambos números no podrá usarse la suma aritmética en complemento a dos por el problema del acarreo.
3. Leer del teclado una secuencia de números enteros positivos hasta que alguno sea 0, almacenándolos en celdas consecutivas a partir de una a determinar, leer otro patrón de teclado y si el último patrón ingresado es \$00 buscar el mínimo de todos ellos y escribirlo en pantalla, o si es \$01 buscar el máximo de todos ellos y hacer lo mismo.

Se evalúa tanto que cada programa se pueda ejecutar con el programa MAQUINA.EXE y devuelva los resultados esperados, como que el desarrollo de los mismos esté debidamente comentado para poder realizar su seguimiento.

### **Conclusiones**

La utilización del simulador para la enseñanza y el aprendizaje de la Arquitectura de Computadoras en la modalidad de trabajo práctico permite al alumno completar las fases de Polya para la creación de un programa, ya que no solamente debe programar el algoritmo en lenguaje de máquina sino que también debe ejecutarlo en la máquina computadora, lo cual le posibilita la verificación de resultados. El alumno no solamente conoce sino también opera.

Los alumnos deben interpretar las entradas y salidas del programa en lenguaje de máquina lo cual hace posible que los conocimientos ya adquiridos puedan ser revisados, controlados, reforzados y profundizados. Se contribuye de esta manera al desarrollo de la capacidad de abstracción.

El uso del programa simulador como herramienta educativa se podría extender a otras materias como Algoritmos y Programación I, desarrollando el módulo completo como una introducción a la programación y a los recursos de un programa y como entrenamiento en la resolución de problemas de lógica binaria.

### **Referencias**

“Enseñanza de Computación en Carreras de Ingeniería”. 2003. Grossi, M. D., Jiménez Rey, E., Servetto, A., Perichinsky, G. IX Congreso Argentino de Ciencias de la Computación. La Plata. 1252 – 1263.

“Introducción a las Ciencias de la Computación”. 1995. Brookshear, G. Addison-Wesley Iberoamericana.

“Simulador de una máquina computadora”. Software desarrollado por el Lic. Servetto para el Curso de Computación (75.01) de la Facultad de Ingeniería de la Universidad de Buenos Aires, 2003.