

## UNA HERRAMIENTA DIDÁCTICA PARA EL ESTUDIO DEL CÓDIGO MÁQUINA EN PROCESADORES DE 64 BITS

S. VICENTE DÍAZ, D. CASCAO CABALLERO, L. MIRÓ AMARANTE,  
M. A. RODRÍGUEZ JÓDAR Y F. DÍAZ DEL RÍO  
Facultad de Informática. Universidad de Sevilla. Avda. Reina Mercedes s/n. 41012.  
SEVILLA. España. Tfno: 954556469. Fax: 954552759. Email: satur@icaro.fie.us.es

*El objetivo principal de esta aplicación es suministrar a los estudiantes una herramienta, con la cual desarrollar pequeñas aplicaciones sobre una arquitectura RISC de 64 bits, sin la necesidad de disponer de la misma. La herramienta desarrollada (SDP64) posee un completo entorno de edición, depuración y emulación de programas en ensamblador sobre la arquitectura Alpha 21064. Por otra parte, se pretende estudiar hasta qué punto un procesador CISC actual con registros de 32 bits (Pentium, Pentium II, Pentium III, etc.), es capaz de emular eficientemente a un procesador RISC de 64 bits.*

### 1. Introducción

Cuando pretendemos estudiar una determinada arquitectura RISC es muy importante aportar una serie de prácticas de laboratorio en las cuales los alumnos asienten sus conocimientos teóricos, pero muchas veces esto no es posible debido a la dificultad de conseguir dicha arquitectura, o bien, por suponer un excesivo gasto (por ejemplo Alpha [1], MIPS, SPARC, etc.), o bien por que la arquitectura no exista en la práctica, como es el caso del DLX<sup>1</sup> [2]. Por este motivo, cuando queremos programar en máquinas de 64 bits, nos vemos obligados a usar emuladores (con posibilidad de depuración) o simuladores.

En este punto tenemos que distinguir entre emulador y simulador. El primero permite la ejecución de un código hecho para una determinada máquina, en otra totalmente diferente (por ejemplo, código Alpha en un procesador x86). En el segundo se tiene un control total sobre la máquina simulada, permitiendo visualizar el comportamiento interno de los registros y la memoria, o también, el comportamiento del procesador en la ejecución de las instrucciones (segmentación, planificación, predicción, etc.), es decir, un simulador puede mostrar cualquier característica interna del procesador.

Las ventajas del uso de emuladores y simuladores son múltiples, por una parte, el coste de su uso es nulo puesto que la mayoría de ellos son "freeware" y corren bajo PC's de bajo precio, y usualmente disponibles en cualquier laboratorio. Por otra parte, permiten estudiar el comportamiento de algunas máquinas didácticas como el DLX, un procesador usado en muchas universidades como referencia de los procesadores RISC y que es germen de los

<sup>1</sup> El DLX es un procesador RISC didáctico. Existen varios emuladores del DLX que emulan perfectamente todas las características del mismo, dos de los más utilizados son el WinDLX [3] y el DLXView.

misimos. Sin embargo, mientras que para el DLX existe una gran variedad de herramientas didácticas, para procesadores de 64 bits no se ha producido una gran difusión.

Algunos simuladores también nos van a permitir cambiar fácilmente la configuración de nuestra arquitectura simulada, añadiéndole memoria, unidades funcionales, algoritmos de planificación dinámica, etc. Con esto no sólo se estudia una arquitectura, sino múltiples configuraciones sobre dicha arquitectura. Otra ventaja de los emuladores es que normalmente están diseñados con herramientas visuales, lo cual permite una gran facilidad de uso y mucha claridad a la hora de visualizar los resultados. También hay que tener en cuenta que, al ser generalmente programas "freeware" que pueden ejecutarse bajo Windows (95/98/NT/2000) o LINUX, los estudiantes pueden realizar pruebas fuera de las horas de prácticas, lo cual les proporciona una gran comodidad en la realización de las mismas.

## 2. La herramienta SDP64

Una herramienta didáctica no debe ser un mero traductor de instrucciones sobre el cual no podemos depurar y monitorizar la ejecución de los programas. El SDP64 o A.D.S! (Sistema de Desarrollo sobre Procesadores de 64 bits) es un entorno de edición, depuración y emulación de programas en ensamblador para microprocesadores de 64 bits. Dado que la arquitectura del código máquina de los procesadores RISC, es similar, el SDP64 puede adaptarse a cualquiera de estas arquitecturas. Para que el SDP64 se adapte a una determinada arquitectura han de cambiarse el módulo de definición del conjunto de instrucciones y el de características de la arquitectura. En nuestro caso se ha adaptado el SDP64 a la arquitectura entera del Alpha 21064, ya que disponemos de una estación de trabajo de ese tipo en nuestros laboratorios. Esto nos va a permitir hacer comparativas entre el código emulado en el SDP64 y el ejecutado en la máquina real.

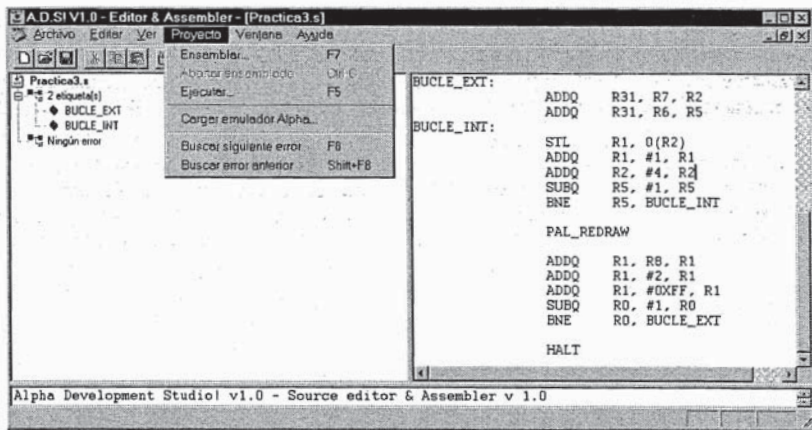


Figura 1: Una sesión abierta del Editor/Ensamblador del SDP64 bajo Windows.

El SDP64 está básicamente formado por los dos módulos siguientes: Editor/Ensamblador y Emulador/Depurador, los cuales se describen a continuación.



### 3. Editor/Ensamblador del SDP64

El Editor/Ensamblador (Figura 1) nos permite crear, editar, ensamblar y corregir cómodamente el código fuente. Incluye una serie de características que nos dotan de una gran libertad a la hora de corregir código y eliminar errores: ventana de etiquetas con búsqueda automática de su posición mediante doble click, listado de errores también con búsqueda automática de posición y con mensaje descriptivo del error, ayuda compuesta de las instrucciones del Alpha, etc. Tras terminar de corregir los errores del código (fichero con extensión .s) se puede ejecutar y depurar directamente el código, ya que la invocación del módulo Emulador/Depurador, carga automáticamente el fichero asociado (extensión .axp).

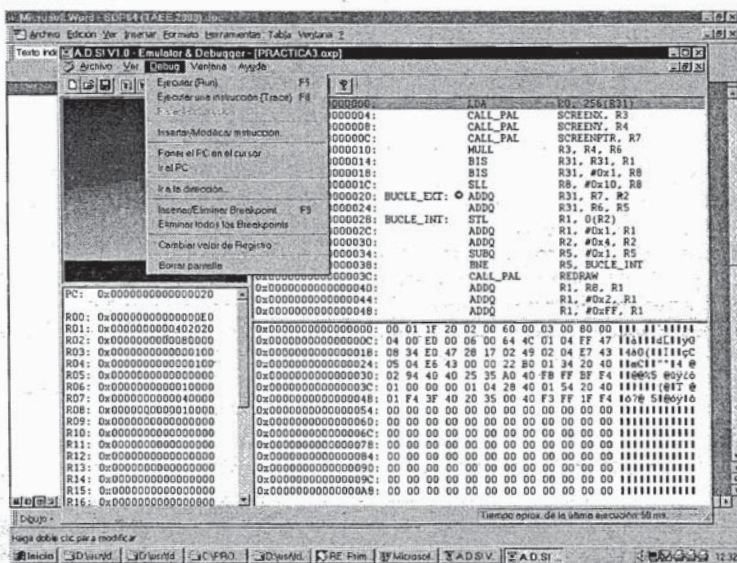


Figura 2: Una sesión abierta del Emulador/Depurador del SDP64 bajo Windows.

### 4. Emulador/Depurador del SDP64

El Emulador/Depurador (Figura 2) nos va a permitir ejecutar y depurar nuestro código 64 bits para Alpha. Incluye características avanzadas como son, puntos de ruptura (breakpoints), un ensamblador/desensamblador de línea, edición de registros y memoria, ejecución paso a paso, etc. Con esto se obtiene una visión detallada del estado de la máquina en cualquier momento de la ejecución de un programa. También se le ha dotado de una pantalla virtual (esquina superior izquierda) a la cual se puede acceder mediante instrucciones PAL de Alpha, y que se puede borrar desde el SDP64. De esta manera el alumno comprueba visualmente los resultados gráficos de la ejecución de su código en tal pantalla virtual (ver Figura 2). En total, el SDP64 soporta la emulación de unas 150 instrucciones de aritmética entera. Para probar la utilidad del SDP64 se han realizado múltiples programas, entre los cuales destaca la implementación del juego "Tetris" para Alpha, cuya salida se realiza sobre la pantalla virtual.

Con todo esto, el alumno puede comprobar las prestaciones de la emulación de un código previamente escrito y depurado por él mismo, y sacar conclusiones de las características de la emulación de procesadores RISC 64 bits sobre plataformas CISC. Para ello se ejecuta el núcleo de su código (sin Entrada/Salida) en la estación Alpha y en el emulador. En concreto, en la asignatura Arquitectura de Sistemas Paralelos I (4º Ingeniería Informática), se realiza una práctica donde los alumnos deben escribir dos versiones del mismo código (un pequeño bucle que recorre una tabla de enteros): una con variables de 64 bits y otra con variables de 32 bits. Se comprueba que, mientras el tiempo de ejecución es idéntico para la estación Alpha; para la emulación sobre los Pentium II del laboratorio, la diferencia de tiempos entre las dos versiones es bastante apreciable.

Otro de los objetivos buscados era medir hasta qué punto se puede emular eficientemente a un RISC de 64 bits sobre un CISC moderno de 32 bits (para obtener la máxima rapidez en la emulación, el núcleo de interpretación de instrucciones Alpha se escribió en ensamblador x86). Se han escrito una serie de "programas de juguete", cuyos resultados para el SDP64 y la estación Alpha real se muestran en la Figura 3. En total la media geométrica nos da un índice relativo de prestaciones entre ambos sistemas de 0.142 (inversa 7.042). Hay que tener en cuenta que el emulador está ejecutando por cada instrucción Alpha, 5 instrucciones Pentium (del código de decodificación), más  $n$  de la propia instrucción emulada.

Algoritmo	Alpha AXP (ms)	PentiumII con SDP64 (ms)	Aceleración
BubbleSort 16Kdatos (caso peor)	899	6.315	0.142
BubbleSort 128Kdatos (caso peor)	56.547	404.957	0.139
256 generaciones del juego de la vida (Conway)	549	3.734	0.146

**Figura 3:** Tiempos de ejecución para programas de prueba seleccionados.

Características de las máquinas: Estación Digital DECpc AXP 150 150Mhz, 64 Mb de RAM, S.O. DEC OSF/1. Ordenador PC Pentium II 233Mhz, 64 Mb RAM, S.O. W' 98.

## 5. Conclusiones

La introducción de la herramienta SDP64 ha tenido una aceptación muy buena por parte del alumnado, ya que les permite escribir, ejecutar y depurar código ensamblador de una máquina de 64 bits de palabra, siendo la mayoría de los emuladores y simuladores didácticos existentes de procesadores de 32 bits. Además, tras la depuración en el SDP64, la ejecución real sobre la estación Alpha motiva al alumno. Por otro lado, dado que la mayoría de procesadores de propósito general serán pronto de 64 bits, el SDP64 podrá ser introducido en el alumnado como su primer contacto con las herramientas de emulación.

## 6. Referencias

- [1] Varios autores. *Alpha Architecture Handbook*, Digital Equipment Corporation. (1992).
- [2] D. A. Patterson y J. L. Hennessy. *Computer Architecture: A Quantitative Approach*, Second Edition. Editorial Morgan Kaufmann Publishers, Inc. (1996).
- [3] Prof. Herbert Grünbacher. *WinDLX: Windows-Deluxe-Simulator*. University of Technology Vienna. Inst. für Technische Informatik. Disponible en muchos servidores Web, por ejemplo en: <http://drago.fie.us.es/software/>