Aprender Arquitectura de Computadores con la herramienta Simula3MS

M. Amor, R. Concheiro, P. González
Departamento de Electrónica y Sistemas
Universidade da Coruña
Campus de Elviña
15071 A Coruña
{margamor,rconcheiro,pglez}@udc.es

M. Bóo, J. A. Lorenzo, D. Piso, R. R. Osorio
Departamento de Electrónica y Computación
Universidade de Santiago de Compostela
Campus Sur
15760 Santiago de Compostela
montserrat.boo@usc.es

Resumen

En este artículo se presenta la experiencia de uso de la herramienta *Simula3MS* en el laboratorio de las asignaturas de Arquitectura de Computadores en los estudios impartidos en la Facultad de Informática de la Universidade da Coruña y la Escuela Técnica Superior de Ingeniería de la Universidade de Santiago de Compostela. El objetivo de las prácticas es ilustrar, mediante un juego de ejercicios realizados sobre el simulador *Simula3MS*, cómo trabaja el procesador del computador. El simulador usa un subconjunto de instrucciones MIPS y varias configuraciones diferentes del procesador que se escogen según el objetivo de los ejercicios prácticos. En el artículo se describen los ejercicios que se proponen a los estudiantes en el laboratorio y se muestra un análisis sobre el impacto que ha tenido el uso del simulador en la actividad docente.

1. Introducción

El objetivo principal de los primeros cursos de Arquitectura de Computadores, en la formación de grado en diversos campos de la ciencia y la ingeniería, es el estudio de los principios básicos cuantitativos del diseño de un computador, que abarca la organización del computador a nivel de unidades funcionales. Uno de los objetivos es proporcionar un conocimiento de las raíces históricas, y por qué los computadores han evolucionado como lo hicieron. Otro objetivo es que comprendan los compromisos involucrados en el diseño del computador, junto con los métodos utilizados para evaluar el rendimiento y realizar estudios comparativos. Por último, y no por ello menos importante, otro objetivo es preparar a los estudiantes para que sean capaces de leer y entender la literatura actual en el campo de la Arquitectura de Computadores, un campo en constante evolución.

Resulta bastante habitual que el funcionamiento del procesador se explique en estos cursos a través de un procesador concreto para facilitar la adquisición de los conceptos, y se utiliza como elemento de apoyo un simulador de ese procesador. Uno de los procesadores más usados en los cursos de Arquitectura de Computadores en muchas universidades y escuelas técnicas es el MIPS, debido a que el repertorio de instrucciones del procesador MIPS es muy simple y reducido. Así, uno de los libros de texto más populares en cursos básicos de arquitectura de computadores es el "Estructura y Diseño de Computadores: interficie cirtuitería/programación" de D. Patterson y J. Hennessy que se basa en el MIPS para explicar el funcionamiento del procesador. *Simula3MS* [3] simula los caminos de datos mostrados en las diferentes secciones de este libro de texto. La herramienta *Simula3MS* se ha diseñado con propósitos pedagógicos y está pensada para ser útil en el aprendizaje de la organización del procesador y la programación en lenguaje ensamblador en los primeros cursos de grado. En este artículo se muestra la experiencia de uso de *Simula3MS*.

El artículo se organiza de la siguiente forma: la sección 2 muestra los principales simuladores pedagógicos del procesador y la motivación a la hora de diseñar e implementar *Simula3MS*; en la sección 3 se presenta la herramienta *Simula3MS*; en la sección 4 se describe la experiencia de uso; y finalmente la sección 5 muestra las conclusiones y algunas ideas para el trabajo futuro.

2. Simuladores Pedagógicos

El uso de simuladores [6, 12] en lugar de procesadores reales resulta bastante común debido los problemas que podría ocasionar programar directamente sobre el procesador.

En la actualidad existen varios simuladores de lenguaje ensamblador con características muy diferentes, los más utilizados son el Spim [5, 9], el DLX [1], el SimuProc [4] y el Simplescalar [2].

La principal ventaja del *Spim* es estar basado en un procesador real. Este simulador permite observar la evolución del segmento de datos (pila y memoria) y de los registros durante la ejecución de las instrucciones. Algunas de las principales carencias de este simulador son la ausencia de una representación gráfica del camino de datos sobre el que se pueda ver la ejecución concreta de cada instrucción, o la imposibilidad de variar la configuración del camino de datos y observar la ejecución de una instrucción por pasos, o ver los posibles efectos de la segmentación sobre el mismo código. Además, no tiene un editor propio, lo que conlleva dificultades para depurar el código o simplemente para hacer cambios en el mismo y tener que cargar el fichero después de cada cambio.

El *DLX* es un simulador de un procesador segmentado que permite ver la representación de la ejecución de cada instrucción sobre el camino de datos. Es uno de los más utilizados en docencia. Tiene una arquitectura carga/almacenamiento y un repertorio de instrucciones de sencilla decodificación. El usuario puede configurar las características del procesador, escogiendo entre varias técnicas de segmentación: básico, Tomasulo y Marcador. La configuración, por parte del usuario, de las características del simulador resultan poco amenas y nada intuitivas. Uno de sus principales inconveniente es que no incluye una representación de la evolución del segmento de datos o de los registros durante la ejecución del código, aunque en su versión para Windows, *WinDLX* incluye una representación del valor de los registros en cada instrucción.

SimuProc tiene un entorno diferenciado donde no aparecen representados ni el segmento de datos ni los registros. Tiene, además, el registro PC (contador de programa) y la memoria situados en direcciones de memoria muy bajas, y su diseño no se basa en ningún procesador real.

El *Simplescalar* es un simulador más orientado a investigación en arquitectura de computadores y no dispone de interfaz gráfica.

Simula3MS es un simulador cuya principal utilidad es su uso pedagógico por lo que el diseño y la implementación de la herramienta ha intentado paliar los defectos detectados en otras herramientas similares. *Simula3MS* fue diseñado para ser fácil de usar y, al mismo tiempo, incluir todos los conceptos importantes relacionados con la arquitectura de la CPU y la programación en lenguaje ensamblador que se explicaban durante las clases magistrales.

3. La herramienta Simula3MS

Simula3MS [3] es un simulador de un procesador configurable, que cuenta con un entorno de trabajo sencillo que permite depurar fácilmente los programas, observar la evolución de la memoria, así como la ejecución de las instrucciones sobre distintos caminos de datos como pueden ser *monociclo*, *multiciclo* y *segmentado*. La presencia de las distintas implementaciones permite observar las diferencias de un mismo código según cuales sean las características del procesador.

Otro punto importante consiste en poder observar la evolución de todos estos 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 su ejecución.

Consta de una primera parte que incluye un editor que permite cargar un programa ya existente en un fichero o editarlo desde cero en la propia herramienta. En esta parte de la herramienta se analiza sintácticamente las instrucciones antes de pasar a la ejecución de las mismas. Una vez analizadas sintácticamente las instrucciones se puede seguir la evolución del segmento de datos, así como de los registros y del resto del camino de datos en la ventana de simulación (ver figura 1).

3.1. Configuración del camino de datos

Como ya se ha mencionado anteriormente, *Simula3MS* permite escoger entre tres configuraciones diferentes del camino de datos: *monociclo*, *multiciclo* y *segmentado*. La configuración monociclo es una implementación simple que usa un solo ciclo de reloj para cada instrucción. Esta aproximación no es práctica pero es apropiada para explicar y entender conceptos básicos del camino de datos del procesador. En la configuración multiciclo cada instrucción se rompe en una serie de pasos donde cada paso consume un ciclo de reloj. Por

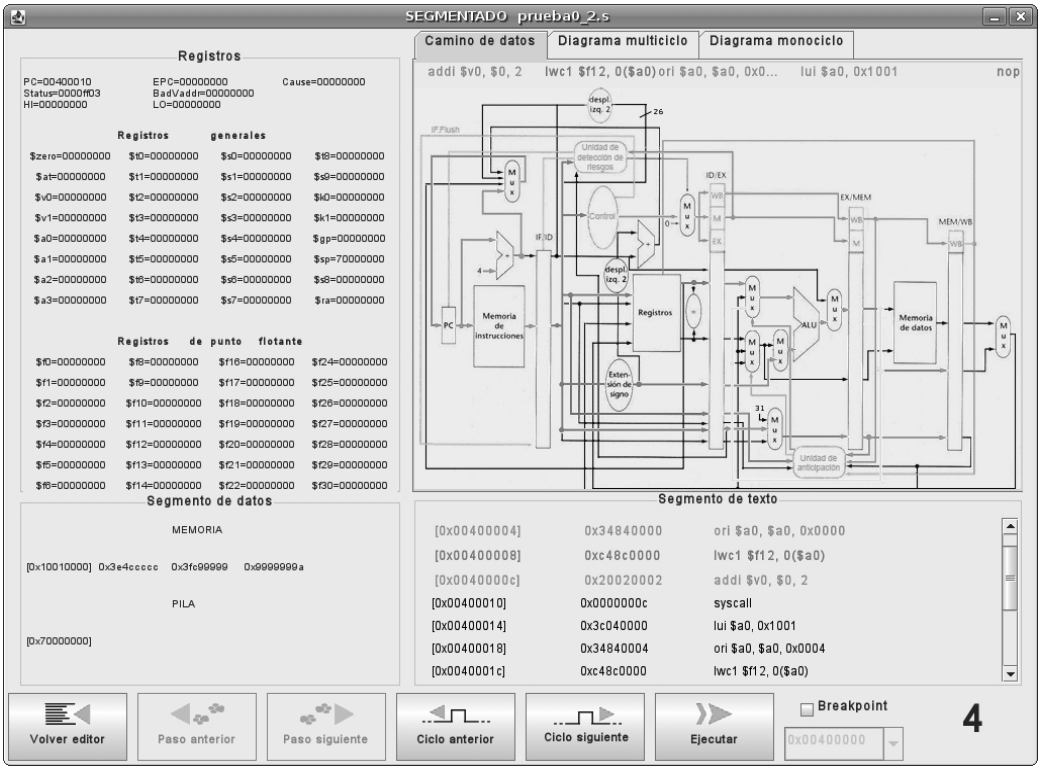


Figura 1: Ventana de simulación, para el procesador segmentado básico

lo tanto, esta configuración permite a diferentes instrucciones consumir diferente número de ciclos. En esta configuración se puede elegir la latencia de las unidades funcionales en punto flotante. La tercera configuración, procesador segmentado, permite seleccionar entre el camino de datos del procesador segmentado básico o dos técnicas de planificación dinámica diferentes, Marcador y algoritmo de Tomasulo. Las técnicas de planificación dinámica se explican más adelante. La configuración básica segmentada en el simulador tiene cinco etapas: búsqueda de la instrucción, decodificación y lectura de operandos, ejecución o cálculo de la dirección, acceso a memoria y postescritura. El simulador implementa una unidad de anticipación que controla los riesgos de datos. En la figura 1 se muestra el camino de datos para el procesador segmentado. Como se puede ver, la simulación resalta en colores diferentes las líneas del camino de datos y las señales de control de cada instrucción que se está ejecutando. Además, cuando se simula el procesador segmentado básico, el estudiante tiene la posibilidad de observar una representación en forma de diagrama en múltiples ciclos de reloj.

En la configuración de procesador segmentado también se puede seleccionar la técnica de procesamiento de los saltos entre dos propuestas: *salto retardado* o *predicción fija*. Con la técnica de salto retardado siempre se ejecuta la siguiente instrucción secuencial tras el salto. Es responsabilidad del compilador, y en este caso del estudiante, encontrar una instrucción válida y útil para situar después del salto.

3.2. Planificación dinámica

Un tema común en cursos más avanzados de Arquitectura de Computadores es el paralelismo a nivel de instrucción (ILP) y cómo explotarlo de forma dinámica.

Una de las mayores limitaciones del camino segmentado básico es que la emisión de las instrucciones y su ejecución se realiza en orden. Las dependencias entre instrucciones cercanas en el cauce dan lugar a riesgos y bloqueos durante la ejecución. En la planificación dinámica que se ha implementado en el simulador, todas las instrucciones pasan la etapa de emisión en orden. Sin embargo, pueden quedarse bloqueadas y ser adelantadas por otras instrucciones en la siguiente etapa entrando en la etapa de ejecución fuera de orden. La ejecución fuera de orden introduce nuevos riesgos que no existían en el camino segmentado básico.

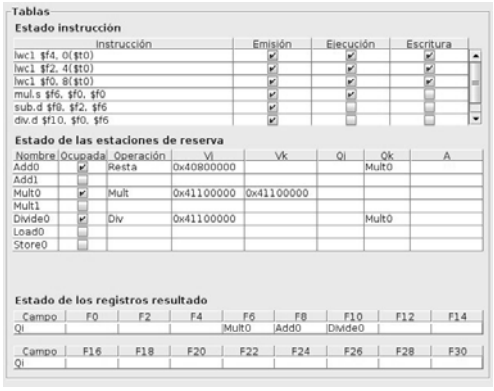
Marcador y el algoritmo de Tomasulo son dos de los esquemas más populares que muestran como se gestionan los riesgos de datos en una planificación dinámica [8]. La última versión de *Simula3MS* permite mostrar la simulación de ambos esquemas (ver figura 2).

3.3. Entrada/Salida

En los cursos introductorios de Arquitectura de Computadores el tratamiento de la Entrada/Salida es a menudo descriptivo y pocas veces incluye discusiones detalladas o ejercicios de laboratorio. En la última versión de *Simula3MS* se ha incluido la posibilidad de simular la Entrada/Salida.

Simula3MS usa Entrada/Salida mapeada en memoria para direccionar dos dispositivos: teclado y pantalla. Se han implementado dos esquemas de control diferentes: *encuesta* e *interrupciones*. En el esquema de encuesta, el procesador comprueba periódicamente el registro de estado del dispositivo de Entrada/Salida para determinar si necesita el servicio del procesador. La desventaja de la encuesta es que el procesador pierde tiempo realizando las funciones relacionadas con la Entrada/Salida. En el esquema basado en interrupciones, el dispositivo emplea señales de interrupción para indicar al procesador que necesita atención.

(a) Marcador



(b) Algoritmo de Tomasulo

Figura 2: Tablas de información usadas en la representación de las técnicas de planificación dinámica

4. Experiencia de uso

Simula3MS se ha usado durante los últimos cinco años en la asignatura de *Estructura de Computadores I* en las titulaciones de Ingeniería Informática e Ingeniería Técnica en Informática de Sistemas en la Universidade da Coruña (UDC) y en los últimos dos años en la asignatura *Estructura de Computadores I* en la titulación de Ingeniería Técnica en Informática de Sistemas en la Universidade de Santiago de Compostela (USC).

La asignatura de Estructura de Computadores I en la UDC es una asignatura cuatrimestral de 7,5 créditos (6 teóricos y 1,5 prácticos) que se imparte en el

primer cuatrimestre del segundo curso. Los alumnos que llegan a esta asignatura han cursado en el segundo cuatrimestre del primer curso la asignatura Tecnología de Computadores, donde se estudian los circuitos lógicos básicos, tanto combinacionales como secuenciales. En el tercer curso los alumnos completarán sus conocimientos básicos de la arquitectura clásica en la asignatura Estructura de Computadores II.

La asignatura de Estructura de Computadores I en la USC es una asignatura cuatrimestral de 6 créditos (3 teóricos y 3 prácticos) impartida en el segundo cuatrimestre del primer curso. Los alumnos cursan previamente, en el primer cuatrimestre, la asignatura de Sistemas Digitales, donde se realiza un primer acercamiento al ordenador como sistema digital. En el segundo curso, los alumnos cursarán la asignatura de Estructura de Computadores II, donde adquirirán un conocimiento más avanzado relativo a la jerarquía de memoria y procesadores superescalares.

El objetivo de ambas asignaturas es el mismo, introducir y analizar los principios básicos de la organización de los computadores. La selección de un libro de texto adecuado para esta materia no presenta ningún problema, ya que existen varios textos excelentes que cubren este campo [10, 11, 7]. En concreto, el libro de texto que se usa fundamentalmente en nuestras asignaturas es el "Estructura y Diseño de Computadores: interficie circuitería/programación" de D. Patterson y J. Hennessy. La dificultad radica en encontrar una serie de ejercicios prácticos para los estudiantes, que les permitan afianzar los conceptos vistos en las clases de teoría.

Tradicionalmente, estas prácticas consistían en proponer a los estudiantes el diseño e implementación en lenguaje ensamblador de un programa en concreto como, por ejemplo, un producto matriz-matriz. Este tipo de prácticas motivaba poco a los alumnos, favorecía el absentismo a las clases de laboratorio (con la excusa de que las podían realizar por su cuenta), y favorecía la copia de prácticas entre ellos o el abandono de la misma antes de finalizarla, especialmente durante los cursos en los que las sesiones prácticas no puntuaban en la evaluación del alumno.

Una vez que dispusimos de la nueva herramienta *Simula3MS* modificamos el planteamiento de las prácticas de nuestras asignaturas. La asistencia de los alumnos al laboratorio pasó a ser obligatoria para poder superar la asignatura. Cada sesión está dedicada a un tema concreto, como pueden ser los saltos condicionales, el paso de parámetros en subrutinas, el uso de la pila, etc. En cada una de las sesiones se entrega a los alumnos un guión de prácticas donde se detallan las instrucciones de la práctica y se indica claramente el objetivo de la misma. Además, estos guiones constan de una serie de cuestiones cortas que deben ir respondiendo durante el desarrollo de la práctica. Este método fomenta en los alumnos una actitud más activa en el laboratorio y permite que saquen mayor partido de cada sesión. Para evaluar a los alumnos, durante el cuatrimestre se les pide que entreguen la respuesta a los ejercicios de los guiones en dos de las sesiones que realizan en el laboratorio. Las respuestas han de entregarlas al finalizar la sesión correspondiente.

La idea es crear ejercicios de laboratorio que fuercen al alumno a reflexionar (antes que simplemente insertar instrucciones y ejecutar código) y evaluar decisiones de diseño. Los ejercicios diseñados se pueden clasificar en una de estas categorías:

- Introducción al lenguaje ensamblador del MIPS. Los ejercicios en esta categoría incluyen estudiar la representación de las instrucciones en el computador, su codificación binaria, el formato de la instrucción y los modos de direccionamiento que utiliza, así como realizar pequeños programas en lenguaje ensamblador, usando subrutinas en las cuales es necesario pasar parámetros y devolver resultados.
- Representación de los datos en el computador. Estos ejercicios incluyen estudiar la representación de los números negativos, determinar los enteros mayores (con y sin signo) que se pueden representar en el computador, estudiar cómo se representan los números reales, determinar los números reales mayores y menores (incluyendo los no normalizados) que se pueden representar en el formato IEEE 754 en simple y doble precisión, y observar problemas de redondeo en algunas operaciones aritméticas.
- Evaluación de diferentes organizaciones del computador. Los estudiantes deben calcular el CPI de un mismo código en diferentes configuraciones, determinar el valor de las señales de control para algunas instrucciones en diferentes configuraciones, modificar un código para

hacer uso del hueco de retardo, identificar riesgos en la ejecución segmentada o planificar el código para mejorar el CPI y comparar las diferencias en rendimiento entre el código original y el modificado.

La acogida por parte de los estudiantes ha sido muy buena, en especial por parte de aquellos que ya tenían experiencia con otros simuladores, o con prácticas de esta materia donde se les pedía simplemente que implementasen y entregasen un determinado programa en lenguaje ensamblador. Además, hemos constatado que los resultados de los examenes de teoría han mejorado con respecto a los cursos anteriores y, en concreto, la asistencia de los alumnos al examen de teoría se ha incrementado espectacularmente. El hecho de que la asistencia al laboratorio sea obligatoria, y de que el esfuerzo continuado que requiere la asignatura durante el curso sea mayor que en los esquemas tradicionales, hace que los estudiantes opten por presentarse a esta asignatura para, por una parte, aprovechar el esfuerzo realizado que les facilita la preparación de la prueba de evaluación teórica y, por otra, evitar tener que repetir las prácticas de laboratorio en el futuro. Por poner un ejemplo, en último curso donde la asistencia al laboratorio no fue obligatoria se presentaron al examen de teoría un 56 % de los estudiantes matriculados, y aprobaron un 44.0 % de los presentados. En el curso 2009/2010, sólo en la convocatoria de febrero, el número de presentados ascendió a un 66 % de los matriculados y aprobaron el examen 54.5 % de los presentados.

Por otra parte, al finalizar cada curso se les ha pasado a los estudiantes un cuestionario de evaluación del planteamiento de las prácticas como ayuda al estudio de los conceptos vistos en teoría y de *Simula3MS* como herramienta pedagógica. En este trabajo mostramos los resultados obtenidos de dicha evaluación en la última edición de las asignaturas. Respondieron al cuestionario un total de 61 estudiantes en la USC y 56 en la UDC, a los que se les pedía que puntuasen cada pregunta en una escala del 1 al 5, siendo 1 totalmente en desacuerdo, y 5 totalmente de acuerdo. En la figura 3 se muestran los resultados del cuestionario, que indican que la mayoría de los alumnos consideran que el simulador les ha ayudado a entender los conceptos vistos en teoría (ver figura 3(a)).

El cuestionario de evaluación también incluía una pregunta sobre qué aspectos de las prácticas se podrían mejorar. Muchas respuestas sugerían que se valorasen todas las sesiones de prácticas. Además los alumnos de la UDC se mostraba partidarios de tener sesiones de dos horas en lugar de sesiones de una hora. Las respuestas a las encuestas realizadas en la UDC indican que los alumnos son poco partidarios del nuevo método de realización y de evaluación de las prácticas, como se refleja en las figuras 3(c)-(d). Estos malos resultados se deben a que las sesiones de una hora resultan demasiado cortas para desarrollar con comodidad las prácticas, y en concreto resultan especialmente cortas para realizar las sesiones de evaluación, teniendo en cuenta el estrés al que se ven sometidos los alumnos cuando se sienten evaluados. Por otro lado, los alumnos de la USC muestran una respuesta más positiva en cuanto al nuevo método de evaluación de las prácticas pero más crítica con la herramienta (ver figura 3(b)-(e)). Estos alumnos disponían de dos horas en cada una de sus sesiones de prácticas. Los resultados de las encuestas se tendrán en cuenta, en la medida de lo posible, para la mejora de la herramienta y para la planificación de los laboratorios en cursos posteriores.

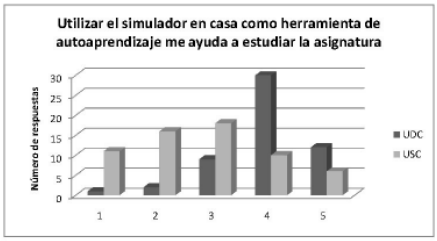
También resulta interesante recoger y analizar los comentarios de los profesores de laboratorio, que corroboran el hecho de que los estudiantes que han usado *Simula3MS* en el laboratorio han mejorado su comprensión del material del curso.

5. Conclusiones

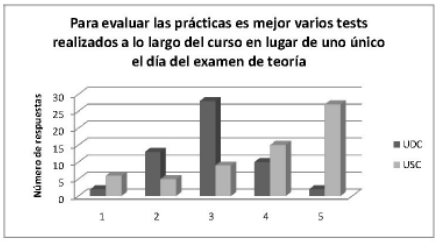
El principal uso para el que fue diseñado *Simula3MS* es pedagógico, por lo que el diseño y la implentación de la herramienta ha intentado paliar las carencias detectadas en otras herramientas similares como Spim o DLX, que se venían usando hasta ahora en nuestras asignaturas.

La característica fundamental de *Simula3MS* es su capacidad para permitir múltiples configuraciones, que permite a los estudiantes comparar diferentes escenarios y les da la oportunidad de evaluar decisiones de diseño. En los primeros cursos de Arquitectura de Computadores es útil poder mostrar a los alumnos el comportamiento de procesadores muy simples, como la implementación del procesador monociclo, y la evolución de esta organización

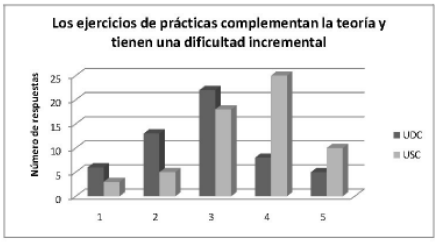
(a)



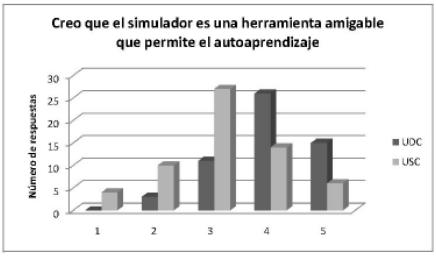
(b)



(c)



(d)



(e)

Figura 3: Resultados de los cuestionarios para la evaluación del *Simula3MS* y el método de evaluación de las prácticas. Escala: 1=Totalmente en desacuerdo, 5=Totalmente de acuerdo.

hacia arquitecturas más realistas como la multiciclo o el procesador segmentado básico.

Simula3MS se usa actualmente en los laboratorios de las asignaturas de Arquitectura de Computadores en los primeros cursos de las titulaciones impartidas en las universidades de A Coruña y Santiago de Compostela. En general, se ha visto una sustancial mejora en los resultados de evaluación y un alto nivel de aceptación por parte de los alumnos de este nuevo planteamiento y evaluación de las prácticas.

Referencias

[1] DLXview. http://cobweb.ecn.purdue.edu/-∼teamaaa/dlxview.

[2] Simplescalar. http://www.simplescalar.com.

[3] Simula3ms. http://simula3ms.des.udc.es.

[4] Simupro. http://simuproc.softonic.com.

[5] R. L. Britton. *MIPS Assembly Languaje Programming*. Prentice Hall, 2004.

[6] S. M. Bruschi et al. Simulation as a tool for computer architecture teaching. In *Summer Computer Simulation Conference*, 1999.

[7] C. Hamacher, Z. Vranesic, and S. Zaky. *Computer Organization*. McGraw-Hill, Inc., 2002.

[8] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 2007.

[9] J. R. Larus. SPIM A MIPS R2000/R3000 Simulator. http://www.cs.wisc.edu/-∼larus/spim.html.

[10] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: the Hardware/Software Interface*. Morgan-Kaufmann, 2005.

[11] W. Stallings. *Computer Organization and Architecture*. Prentice Hall, 2000.

[12] G. S. Wolffe, W. Yurcik, H. Osborne, and M. A. Holliday. Teaching computer organization/architecture with limited resources using simulators. *ACM SIGCSE*, 2002.