

Una herramienta para el aprendizaje del procesador DLX

Perna, Juan I.; Grosso, Alejandro L.; Berón, Mario M.

Departamento de Informática

Facultad de Ciencias Físico, Matemáticas y Naturales

Universidad Nacional de San Luis – Argentina

e-mail: {jiperna, agrosso, mberon}@unsl.edu.ar

Fax: 54-2652-430224

RESUMEN

En este artículo se presenta una herramienta para el aprendizaje del procesador DLX. Entre los objetivos principales de la misma se encuentran: facilitar el entendimiento del funcionamiento del procesador de estudio, reforzar los conceptos subyacentes en la programación en lenguaje ensamblador, y servir de apoyo en las clases teóricas y prácticas de la materia Arquitectura del Procesador I de la licenciatura y el profesorado en Ciencias de la Computación. De esta forma se pretende brindar diferentes vías para que el alumno adquiera los conceptos básicos de la materia.

La herramienta proporciona un ambiente de programación atractivo y simple, es multiplataforma e incorpora un simulador, ensamblador y cargador, con las funcionalidades esenciales para el desarrollo de programas a nivel de código de máquina.

Este software fue diseñado teniendo en cuenta las herramientas existentes como así también la experiencia docente adquirida a través del seguimiento de las clases teóricas y prácticas de la materia Arquitectura del Procesador I.

Palabras Clave: Arquitectura del Procesador, Procesador, Aprendizaje Significativo, Relación triádica.

1. INTRODUCCION

La enseñanza de las arquitecturas del procesador [7] [8] [18] es de principal importancia ya que permite tener un conocimiento acabado de las potencialidades de la máquina que se está usando, siendo esto de valor a la hora de construir software de alta calidad.

A través de la experiencia adquirida después de varios años del dictado de la materia, se pudo comprobar que los conceptos básicos de esta temática son difíciles de comprender por el alumno que se inicia en el estudio de las ciencias de la computación. Atentos a estas observaciones, el equipo de cátedra se abocó al diseño de teorías, prácticas y herramientas que disminuyan estas dificultades.

Se puede decir que los contenidos de la materia Arquitectura del Procesador I pueden ser divididos en dos grandes grupos: circuitos digitales y programación en lenguaje ensamblador. Como herramienta didáctica para el primer grupo temático se diseñó un conjunto de circuitos básicos los cuales sirven de base para la construcción de otros más complejos planteados en las prácticas de laboratorio de la materia. Mientras que la herramienta presentada en este artículo servirá como apoyo para el segundo grupo de contenidos.

La presente herramienta se desarrolló considerando que es necesario contar con instrumentos capaces de lograr en el alumno un aprendizaje tanto de tipo significativo [6], como memorístico. Se

sabe que siempre es conveniente lograr un aprendizaje significativo que logre correlacionar los nuevos conceptos con los ya adquiridos, pero también es conveniente lograr aprendizajes memorísticos ya que recordar fórmulas matemáticas, instrucciones de máquinas, etc. es indispensable.

Es necesario también considerar que la herramienta debe poseer un diseño que logre en el alumno una motivación que tienda a elevar su autoestima, así como incentivar su gusto por el conocimiento como medio de superación personal.

El hecho de que el proceso de enseñanza-aprendizaje sea significativo tiene su base en el bagaje de conocimientos previos (“Averigua lo que el alumno ya sabe y enseña consecuentemente”. Ausubel) que el alumno posee. Basados en la calidad de éstos y en el método de enseñanza aplicado, es como los nuevos conocimientos serán incorporados a la estructura cognitiva del alumno.

No solo la herramienta debe considerar éstos aspectos relacionados con el alumno sino que también debe permitir la interacción entre los docentes que aportan sus conocimientos, ya que cada uno de ellos, a su vez, se enriquece con los conocimientos del otro. Es deseable esperar que esta interacción contribuya a fortalecer la relación triádica Docente-Alumno-Conocimiento [5] [13].

Esta relación permite analizar el proceso de enseñanza aprendizaje desde diferentes aristas: Docente-Conocimiento, Docente-Alumno, Alumno-Conocimiento y la relación en sí misma. Cada una de ellas influye a la hora de impartir-recibir conceptos. Este trabajo propone una herramienta de aprendizaje que tienda a fortalecer la relación Alumno-Conocimiento.

Se ha observado que la relación alumno-conocimiento, en ciertas ocasiones se ve deteriorada por dificultades en el entendimiento del concepto en sí mismo, como así también por las herramientas de apoyo en el aprendizaje [2] [10] [16]. Vale la pena destacar que las herramientas existentes con este fin presentan características no deseables como lo son: dificultad de uso, interfaces poco desarrolladas, presentación obscura de conceptos, ligadura a una plataforma en particular, ayudas escasas o poca descriptivas, escaso fomento del descubrimiento, concepción no basada en teorías del aprendizaje, etc. Esto, generalmente, lleva a que el alumno se centre más en conocer el funcionamiento del software utilizado que en entender el concepto; en este caso la herramienta utilizada se convierte en un distractor para la adquisición de conocimientos, lo cual contribuye a que el alumno estudie por la necesidad de aprobar el curso [14]. Ante las circunstancias planteadas anteriormente, el equipo de cátedra se encuentra abocado a la construcción de herramientas que brinden una solución a esta problemática y, en particular, el Ambiente de Programación presentado en este artículo es una de ellas.

El artículo está organizado como sigue. En la sección 2 se presenta el contexto (problemáticas, teorías, prácticas) de la materia Arquitectura del Procesador I (donde se utilizará la herramienta). En la sección 3 se describe el software con sus funcionalidades principales y se hace énfasis en el aporte realizado por el mismo para la enseñanza del procesador DLX [9]. Por último, en la sección 4, se presentan las conclusiones y visión de futuro.

2. EL CONTEXTO

La materia Arquitectura del Procesador I es dictada en el primer cuatrimestre del segundo año de la carrera Licenciatura y Profesorado en Ciencias de la Computación en la Universidad Nacional de San Luis.

Esta asignatura, junto con la materia Arquitectura del Procesador II, es donde se imparten conocimientos avanzados de procesadores y conforman la única posibilidad (con excepción de las especializaciones) que el alumno tiene de conocer (a nivel de hardware y programación a bajo

nivel) los conceptos troncales de los procesadores actuales. Por esta razón, el equipo de área considera prioritario que los conceptos sean adquiridos por los alumnos de forma significativa [12] [17], lo cual posibilita la transferencia de los mismos a situaciones reales con la menor cantidad de inconvenientes posibles.

Se toma como caso de estudio el procesador DLX porque es un procesador RISC que son la tendencia del momento. Esto se debe a que las tecnologías actuales optan por trasladar hacia los compiladores la tarea de proveer soporte para las características alto nivel de los lenguajes en lugar de su implementación en las arquitecturas. De esta forma, se logra una simplificación en la lógica de los procesadores que redundará en una reducción de costos y una mayor eficiencia del producto final.

Se ha observado que los alumnos llegan a cursar la materia con escasos conocimientos en programación y limitadas experiencias en el manejo de software para el desarrollo de aplicaciones. Esto tiene un impacto negativo en el desarrollo de la materia ya que, en ciertas ocasiones, se necesita retomar las teorías o bien extender la duración de las clases prácticas para que el alumno disponga del tiempo de reflexión necesario para entender los conceptos.

El número de alumnos que cursan la materia Arquitectura del Procesador I oscila entre 120 y 150. La totalidad de los mismos concurren a las clases teóricas, mientras que para el caso de las prácticas de aula son divididos en tres comisiones, lo que lleva a un número de 40 a 50 alumnos por grupo. El caso de las clases prácticas de laboratorio (entiéndase por estas aquellas que implican el desarrollo de prácticos de máquina o construcción de circuitos) la situación es un tanto más compleja porque la cantidad de ordenadores disponibles para este fin es aproximadamente de 30, lo cual lleva a distribuir a los alumnos en dos comisiones de aproximadamente 60 alumnos (es decir dos por máquina). Como dificultad adicional, en esta última situación, se presentan ocasiones en que el software que se utiliza no está pensado para ejecutar en las plataformas disponibles en el laboratorio, con lo cual el equipo de cátedra se ve en la necesidad de recurrir a otros programas de características similares que puedan ejecutar en los equipos del laboratorio (aún si el mismo no presenta los requerimientos pedagógicos necesarios para la adquisición de este tipo de conocimientos).

Las teorías y las prácticas de la materia están orientadas a cumplir con los siguientes objetivos:

- Obtener el conocimiento básico para el posterior entendimiento de las arquitecturas existentes.
- Operar con todas las variedades de tipos de datos que se pueden presentar en el estudio del funcionamiento de los distintos procesadores.
- Aprender los elementos discretos que se utilizan en el diseño y construcción de hardware.
- Dominar la programación en lenguaje ensamblador.
- Conocer técnicas de traducción de sentencias de alto nivel en instrucciones de lenguaje ensamblador.
- Interiorizarse de las diferentes convenciones para la traducción de subrutinas y las distintas formas de pasaje de parámetros.
- Adquirir los conocimientos inherentes a la programación de la entrada – salida programada y con interrupciones.

2.1. Teorías

En esta sección se describen de forma sintética, los principales contenidos teóricos desarrollados durante el dictado de la materia Arquitectura del Procesador I. Los mismos, son estructuras de cómo se muestra en la tabla siguiente.

Sistemas Numéricos
Introducción a la representación de números en distintos sistemas numéricos y en diferentes bases con sus respectivas operaciones aritméticas.
Circuitos Digitales
Funcionamiento y construcción de componentes básicas de un procesador: flip-flops, registros, contadores, desplazadores, sumadores, etc. Diferentes organizaciones de memorias.
Organización Básica de una Computadora
Se describen a nivel general las distintas componentes de un procesador: unidad de control, ALU, etc. y sus interconexiones; instrucciones básicas de un procesador genérico, sus modos de direccionamiento, etc.
Lenguaje Ensamblador, Ensamblador y lenguaje de programación
Se conceptualiza el lenguaje ensamblador. Se explica el funcionamiento de un ensamblador, detallando los tiempos de resolución de las operaciones, etc..
Entrada-Salida
Módulos de entrada/salida: interfaz con la CPU e interfaz con el dispositivo. Protocolos de entrada/salida. Entrada/Salida programada. Organización de la entrada/salida: dedicada y embebida en memoria. Ejemplos de la entrada/salida en el procesador DLX.
Interrupciones
Tipos de excepciones. Estructura lógica del sistema de interrupción. Prioridades y habilitación de interrupciones. Acceso directo a memoria. Robo de ciclo.

2.2. Las Prácticas

Las prácticas están divididas en áulicas y de laboratorio. La finalidad de las mismas es fomentar la reflexión y transferencia de los contenidos impartidos en las teorías a través de la resolución de ejercicios prácticos, como así también con la construcción de circuitos y programas en el laboratorio.

En la siguiente tabla se muestran los prácticos de aula y de laboratorio de la materia.

Sistemas Numéricos
Representación de números en distintas bases, operaciones aritméticas, representación de números decimales.
Circuitos Digitales
Comprensión del funcionamiento de las compuertas lógicas. Construcción de componentes basados en lógicas combinacionales: sumadores, substractores, etc. Construcción de componentes basados en lógicas secuenciales: registros, contadores ascendentes y descendentes, desplazadores, memorias, etc.
Programación básica en lenguaje ensamblador del DLX
El práctico consiste en resolver problemas simples tales como: carga y almacenamiento de datos, sumas, restas, desplazamientos, etc. con el fin de que el alumno se relacione con el repertorio de instrucciones del procesador.

Programación avanzada en el lenguaje ensamblador del DLX: Modos de Direccionamiento
Los ejercicios se centran en el acceso a estructuras de datos tales como: arreglos, listas, pilas, filas, etc., que inducen al alumno a la utilización de los diferentes modos de direccionamientos (reales o conceptuales) del procesador.
Programación avanzada en el lenguaje ensamblador del DLX: Subrutinas y Parámetros
El práctico introduce las instrucciones de llamada y retorno de subrutinas provistas por el procesador y plantea una serie de ejercicios que deben ser resueltos de acuerdo a las convenciones y tipo (valor y dirección) de pasaje de parámetros (en pila, área asociada a la subrutina, área asociada al llamador, en línea, etc.) vistas en teoría.
Programación en lenguaje ensamblador del DLX: Traducción a Lenguaje de Máquina
Los ejercicios se centran en entender el proceso que se realiza para transformar los mnemónicos de las instrucciones a su correspondiente cadenas dígitos binarios.
Entrada-Salida e Interrupciones
Se introducen las distintas formas en que los dispositivos pueden ser manejados por el procesador central, las instrucciones necesarias para la entrada-salida e interrupciones y se resuelven aplicaciones tales como: lecturas, escrituras, con entrada-salida controlada por la CPU y también con interrupciones.

3. LA HERRAMIENTA PROPUESTA

En esta sección se enmarca el ambiente de programación [3] [4] [11] desarrollado en una categoría de software educativo y se describen las funcionalidad principales del mismo. Así mismo, se resaltan los aportes de la herramienta como apoyo para la enseñanza de la arquitectura del procesador DLX.

3.1 Tipo de Software Educativo

Se puede decir que el software educativo puede ser categorizado en: software instructivo o software constructivo. El primero de ellos intenta que el alumno aprenda el contenido utilizando como estrategias el enlace de conocimientos que parten de lo particular hacia lo general. La presentación de los conceptos es llevada a cabo por la computadora la cual presenta preguntas que el alumno debe responder, el cual a su vez recibe como respuesta una estimación del grado de correctitud de su solución. Podríamos decir, que este tipo de software pretende llevar a cabo un aprendizaje dirigido.

Por otra parte, el software constructivo fomenta en el alumno un aprendizaje por descubrimiento y el desarrollo de la intuición. El software de este tipo propone un ambiente para la exploración y el descubrimiento. Entre las características principales se puede mencionar la tendencia a propiciar la concreción de las ideas como así también el establecimiento de esquemas conceptuales. Como ejemplo de este tipo de software se pueden mencionar los programas de simulación.

Se puede decir que la herramienta presentada en este artículo es un software de tipo constructivo ya que el mismo no presenta una guía estructurada de conceptos básicos interrelacionados sino más bien provee un ambiente en donde el alumno puede: experimentar/visualizar [1] las diferentes componentes (lo cual posibilita que pueda validar o no sus hipótesis); simular el comportamiento del procesador a través de la ejecución de programas y sentirse motivado a descubrir las distintas funcionalidades de la herramienta (dada la flexibilidad e interfaz usuario-máquina provista).

Se piensa que este trabajo es una buena aproximación, para el alumno principiante, hacia el uso de herramientas para el desarrollo de aplicaciones.

3.2 La Herramienta y sus Aportes en la Enseñanza del Procesador DLX

Los aportes principales de la herramienta apuntan la familiarización en el uso de sistemas para construir software de aplicación y en el aprendizaje de la programación en lenguaje ensamblador del procesador DLX. Como efecto colateral, el sistema permite estudiar sistemas numéricos, así como codificar instrucciones y configurar un ambiente [15] personalizado de entrada salida.

Por otra parte, es indispensable destacar que si bien ésta es una herramienta pedagógica, con ella se proveen funcionalidades similares a aquellas que se encuentran en los entornos de desarrollo de uso profesional. Sin embargo, las mismas se presentan de forma simplificada y modificadas para tener un mayor impacto visual fomentando de esta forma la experimentación por parte del usuario novato.

Una visión general de la aplicación se muestra en la siguiente figura 1.

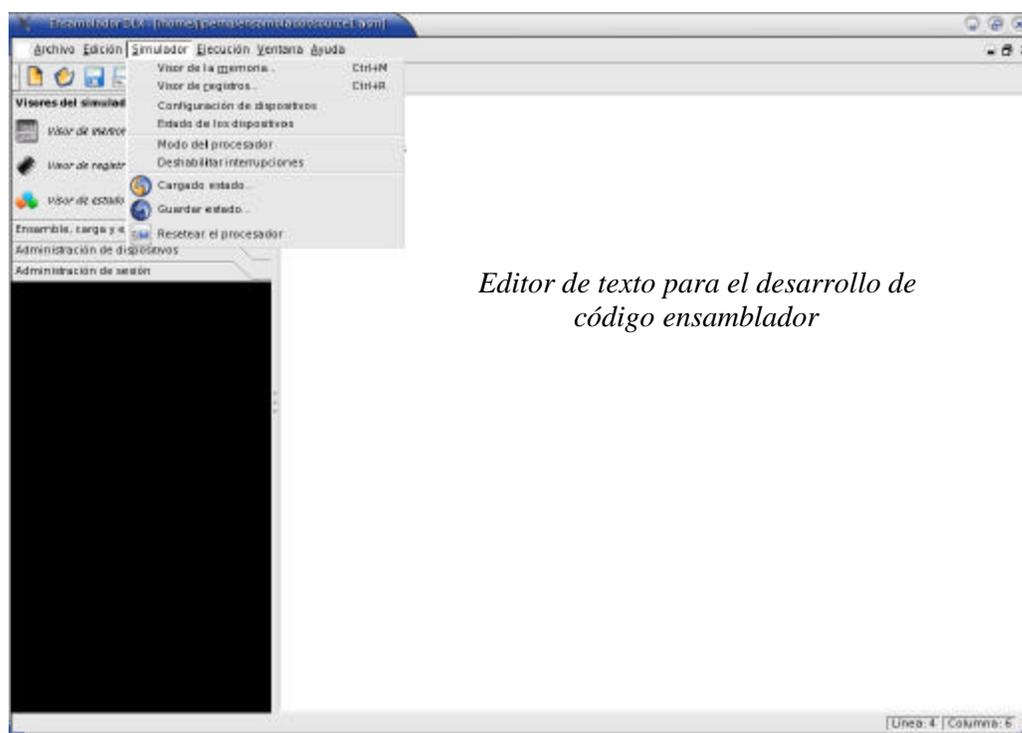


Figura 1: Entorno de desarrollo de la herramienta

Como se puede observar, el ambiente proporciona una barra de menú y una barra de herramientas, ambos con íconos que describen claramente las funcionalidades asociadas.

Es importante destacar que cuando se programa en lenguaje ensamblador, es necesario (con propósitos de depuración) poder analizar el comportamiento del programa a través de la visualización del estado de los distintos componentes del procesador. Para esto, la herramienta provee visores de:

- La memoria : muestra información del contenido de la memoria en distintas bases (binario, hexadecimal) así como en formato desensamblado. Por otra parte, se brinda la posibilidad de señalar la palabra de memoria apuntada actualmente por el contador de programa y de modificar el contenido de cualquier dirección de la memoria. Así mismo, es posible seleccionar la dirección que se desea visualizar (tanto en forma absoluta como en forma relativa dentro de dos espacios de direcciones posibles: del sistema y del usuario).



Figura 2: Visor de memoria

- Registros del procesador: brinda la posibilidad de visualizar y editar el contenido de los registros, tanto los visibles al usuario (propósito general) como los registros especiales. Como característica sobresaliente, se proporcionan funcionalidades de drag&drop (ver figura 3).

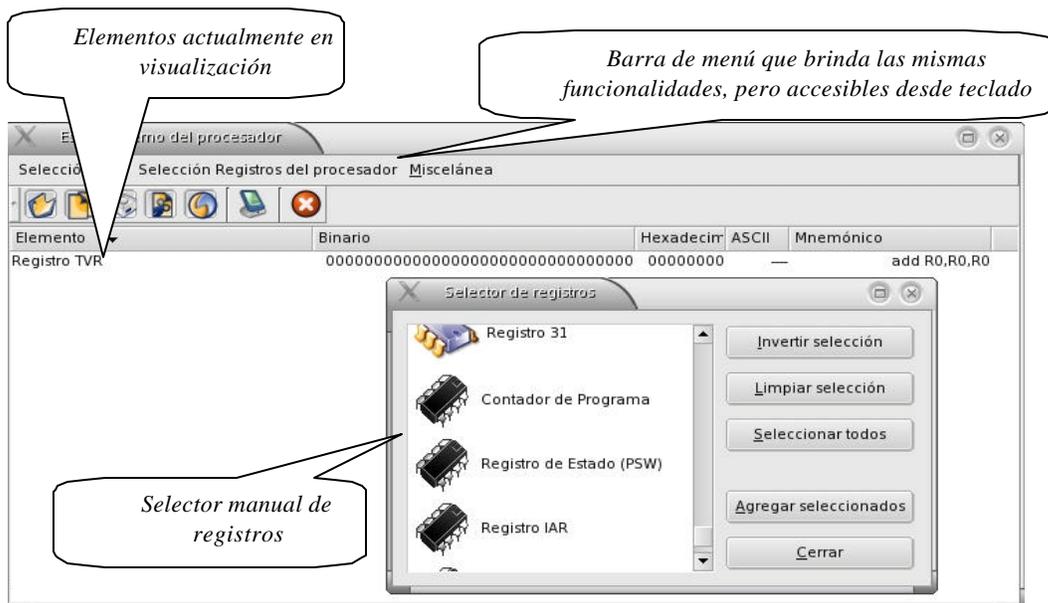


Figura 3: Visor de los registros del procesador

Por otra parte, durante el proceso de ejecución es posible:

- Ejecutar paso a paso

- Ejecución total
- Ejecución hasta llamado a subrutina
- Ejecución hasta un determinado valor del contador de programa o un número fijo de instrucciones hacia adelante.

A continuación se muestra la ventana que proporciona todas las funcionalidades descritas anteriormente.



Figura 4: Sistema de ejecución

El sistema de interrupciones y de administración de dispositivos de entrada/salida se muestra en la sección siguiente mediante un ejemplo de aplicación.

3.4. Concreción de Ideas

Entre los prácticos avanzados que se encuentran en la materia Arquitectura del Procesador I se puede mencionar la programación de los dispositivos de entrada - salida e interrupciones. Entre las dificultades observadas en el desarrollo de los prácticos, surgen como significativos la abstracción de los conceptos y la complejidad introducida por la inclusión de las interrupciones en el funcionamiento del procesador. Esto lleva a que el alumno no consiga entender la forma de solucionar los problemas planteados.

La herramienta, al permitir la incorporación de dispositivos y su administración mediante entrada - salida con interrupciones, concretiza estas ideas poco tangibles del modelo teórico, siendo ésta una característica saliente, no encontrada en los demás simuladores para este procesador. Esto fomenta que el alumno plantee sus propios problemas, formule hipótesis acerca del comportamiento del programa que lo resuelve y las pueda verificar a través de la ejecución simulada del programa.

A continuación se presenta un ejemplo de cómo una situación compleja puede ser manejada en forma sencilla y atrayente mediante el uso de la herramienta: En un entorno industrial, se plantea la necesidad de controlar mediante un procesador DLX el nivel de agua en un tanque contenedor. El mismo cuenta con dos fotocélulas, una que se activa cuando el nivel del agua esta por debajo de un

umbral mínimo, mientras que la otra se enciende cuando el agua supera un nivel máximo preestablecido. Por otra parte, se cuenta con un dispositivo que administra el ingreso de agua (que puede encontrarse en uno de dos estados: abierto o cerrado). Se pretende construir un programa en el lenguaje ensamblador del procesador DLX que administre el ingreso de agua dentro del tanque basado en los datos provistos por las fotocélulas.

La herramienta puede ser adaptada para contemplar esta situación mediante la incorporación de tres dispositivos mapeados en memoria utilizando el administrador de dispositivos. Esta utilidad de la aplicación permite incorporar dispositivos tanto de entrada como de salida y especificar tanto las direcciones de las puertas de datos y control y estado del mismo como la línea de interrupción mediante la cual se comunica con la CPU.

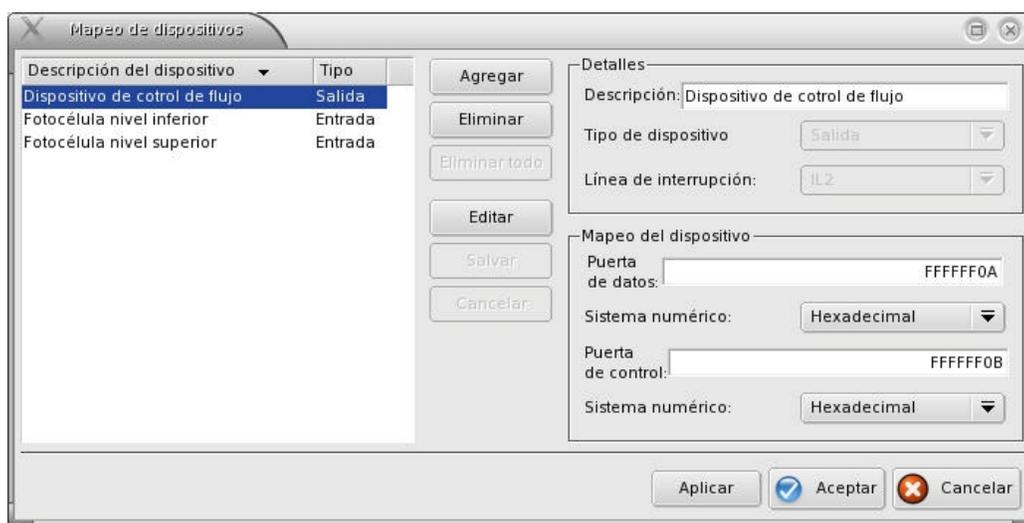


Figura 5: Mapeo de dispositivos en memoria

Una vez configurados los dispositivos (tal como se muestra en la figura 5), es posible retornar al entorno de desarrollo mostrado en la figura 1 en donde se puede escribir el programa que resuelva el ejercicio y utilizar todas las funcionalidades de simulación.

4. CONCLUSIONES Y VISIÓN DE FUTURO

En este artículo se presentó una herramienta que sirve como apoyo para el aprendizaje del procesador DLX (Deluxe). La aplicación fue construida teniendo en cuenta los softwares existentes y la experiencia adquirida a través, de la participación docente, en las clases teóricas y prácticas de la materia Arquitectura del Procesador I. Se pudo comprobar que la construcción de un software de estas características exige un trabajo interdisciplinario y un arduo trabajo de programación que implica la relación de diferentes materias de la carrera Licenciatura en Ciencias de la Computación. Por otra parte, la complejidad de la herramienta hizo necesario la aplicación de métodos y técnicas ingenieriles para su desarrollo, siendo esto de valor a la hora de documentar la aplicación.

Se mostró con un ejemplo como el alumno iniciado en el estudio de las Ciencias de la Computación, a través del uso de la herramienta, puede concretizar situaciones que son demasiado complejas y abstractas. Además se describió, como la herramienta fomenta el aprendizaje por descubrimiento y se destacó que la misma es una buena aproximación hacia el uso de software profesional para el desarrollo de aplicaciones.

Como visión de futuro se pretende utilizar esta aplicación en el análisis del desempeño de distintas versiones del DLX. Además se proyecta construir ambientes de programación para diferentes procesadores, lo cual permitirá ampliar la visión y perspectiva de la materia.

5. BIBLIOGRAFIA

- [1] Ardenghi, J.; Castro, S.; Estévez, E.; Fillottrani, P.; Martig, S.; Sánchez, M.. “*Arquitectura de Sistemas de Visualización*”. WICC 2003. Tandil. Argentina. 2003.
- [2] Bentolila, S.; Clavijo, P.. “*La Computadora Como Mediador Simbólico de Aprendizajes Escolares: Análisis y Reflexiones desde una Lectura Vigotskiana del Problema*”. Año: 1996.
- [3] Berón, M. ; Tertusio, R. ; Blanco, J.. “*Una Aproximación hacia la Construcción de Sitios Web Didácticos*”. WICC 2003. Tandil. Argentina. 2003.
- [4] Chapov, O.; Pizarro, R.. “*Aplicación de un entorno hipermedial para la integración de conceptos matemáticos*”. CACIC 2003. La Plata. Argentinian. 2003.
- [5] Del Rosal, O.. “*La Comunicación Docente-Discente*”. Ed. Plus Ultra. Año:1993.
- [6] Gardner H.. “*Estructuras de la Mente*”. Ed: Basic Book. Año: 1983.
- [7] Gear, William C. “*Computer Organization and Programing*”. Ed. Mc. Graw Hill.
- [8] Hayes J. “*Computer Architecture and Organization*”. Ed. Mc. Graw Hill. 1988
- [9] Hennessy, John L.; Patterson, David. “*Computer Architecture: A Quantitative Approach*”. Segunda Edición. Ed. Morgan and Kaufmann. 1990.
- [10] Martí, E.. “*Aprender con Ordenadores en la Escuela*”. Ed: ICE – HORSORI. Año: 1992.
- [11] Martig, S.; Señas, P.. “*Herramientas y Estrategias Visuales para Ambientes de Aprendizaje Computacionales en la Educación no Presencial*”. WICC 2003. Tandil. Argentina. 2003.
- [12] Piaget, J.. “*Investigaciones sobre la Abstracción Reflexionante*”. Vol I y II. Ed. Huemul. Año: 1979.
- [13] Pichon-Riviere, E. “*Teoría del Vínculo*”. Ed: Nueva Visión. Año: 1995.
- [14] Quiroga, A.. “*Matrices de Aprendizaje*”. Ediciones Cinco. Año: 1991
- [15] Señas, P.. “*Ambientes de Aprendizaje Computacionales para la Educación en sus Diferentes Modalidades*”. WICC 2003. Tandil. Argentina, 2003.
- [16] Solomon. C.. “*Entornos de Aprendizajes con Ordenadores*”. Ed: Ediciones Paidos; Año: 1995.
- [17] Vargas Castillo, C.. “*¿Instructivismo o constructivismo?: Guía multimedia para el desarrollo de software educativo*”. CACIC 2003. La Plata. Argentina. 2003.
- [18] Wakerley, John. F. “*Microcomputer Architecture and Programming*”. Ed. John Wiley and Sons. 1981.