

## ECODE: Entorno Integrado de Desarrollo para CODE-2

Antonio Martínez   Alberto Prieto   Héctor Pomares   Pedro Castillo

Departamento de Arquitectura y Tecnología de Computadores  
E.T.S.I. Informática  
Universidad de Granada  
E-18071 Granada. e-mail: [aprieto@ugr.es](mailto:aprieto@ugr.es)

### Resumen

En este artículo, se describe el funcionamiento y las características más sobresalientes de ECODE, un entorno integrado de desarrollo (IDE) para el Computador Didáctico Elemental CODE-2. ECODE permite editar, ensamblar y depurar programas escritos en lenguaje ensamblador de CODE-2, al modo tradicional, además de emular la ejecución de programas en lenguaje máquina. El entorno puede trabajar en dos modos: modo real y modo ampliado, con el que se visualizan y monitorizan los contenidos de cualquier elemento interno del computador, pudiendo en cualquier instante interactuar con ellos, modificándolos directamente.

### 1. Introducción

CODE-2, es la segunda versión de un Computador Didáctico Elemental diseñado en el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada [1,2]. Se trata de un computador de tipo von Neumann, que contiene las unidades típicas de este modelo: entradas, salidas, unidad de control, unidad de procesamiento de datos y memoria para datos e instrucciones. La filosofía de concepción y diseño de este computador, es eminentemente pedagógica, y está completamente descrito a nivel de lenguaje máquina, lenguaje ensamblador y micromáquina en [2]. La longitud de palabra de CODE-2 es de 16 bits, y dispone de tan sólo 16 instrucciones máquina. Los elementos a los que se tiene acceso directamente desde el nivel de descripción de lenguaje máquina son:

- Un banco de 16 registros (r0 a rF).

- Unidad Aritmético Lógica (ALU), que realiza la operaciones que más adelante detallaremos. La ALU tiene asociados biestables indicadores, que funcionan de forma usual, de acuerdo con los resultados obtenidos en ella: biestable de cero (Z), de signo (S), de acarreo (C) y de desbordamiento (V).
- Memoria Principal: de  $2^{16}$  = 64Kpalabras de 16 bits (128 Kbytes), direccionable por palabras.
- Puertos de entrada y salida: Se admiten 256 puertos de entrada (IP00...IPFF) y otros 256 puertos de salida (OP00...OPFF).

El lenguaje máquina de CODE-2 se compone de 16 instrucciones, todas ellas de 16 bits, y son las siguientes:

- *Transferencia de datos*: Carga en un registro de un contenido de memoria (*LD*), almacenamiento del contenido de un registro en memoria (*ST*), carga inmediata en registro (byte alto, *LLI*, o byte bajo, *LHI*), entrada (*IN*) y salida (*OUT*).
- *Aritmético-Lógicas*: Suma (*ADDS*), resta (*SUBS*), NAND (*NAND*), y desplazamientos lógicos y aritméticos (*SHL*, *SHR*, *SHRA*).
- *Bifurcaciones* incondicionales y condicionales: salto (*B-*), llamada a subrutina (*CALL-*), retorno (*RET*) y
- *Parar* (*HALT*).

A continuación se presenta la primera versión de ECODE (Entorno CODE), un conjunto integrado de herramientas para desarrollar programas en lenguaje ensamblador (definido según el estándar IEEE 694 [4]) o máquina, depurarlos, y analizar con detalle el comportamiento dinámico de CODE-2 al ejecutar programas en código máquina.

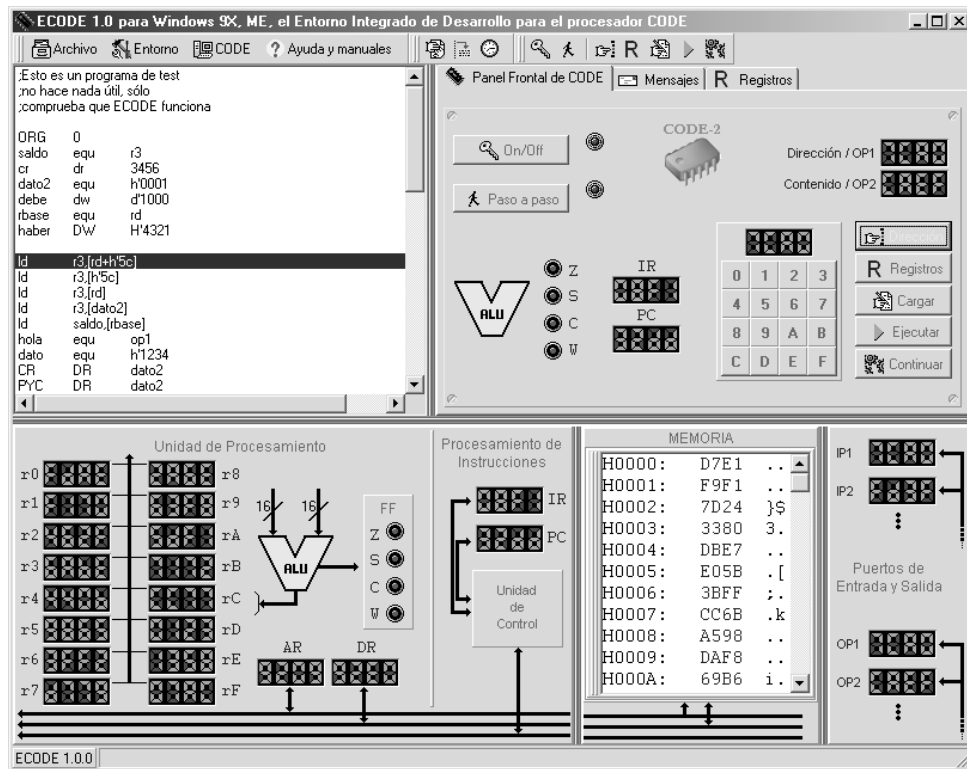


Figura 1. Imagen de la interfaz gráfica de ECODE.

## 2. Especificaciones de diseño de ECODE

ECODE es un Entorno Integrado de Desarrollo (IDE) para el computador CODE-2 que aúna diversas herramientas y programas diseñados con el propósito de conformar un entorno didáctico para el aprendizaje del funcionamiento de un computador básico.

A tal efecto, se definen en ECODE dos modos de funcionamiento, el modo real y el modo ampliado [3]:

- En el *modo real* se emula exactamente el comportamiento de CODE-2; es decir, el alumno sólo puede interactuar con la herramienta, de forma similar a como lo haría con un prototipo físico del computador. Concretamente, sólo se tiene acceso en este modo al panel frontal de CODE-2 descrito en

[2] que dispone de teclado (puerto *IP1*) para introducción de información en hexadecimal, dos puertos de salida (*OP1*, *OP2*), teclado de órdenes (cargar, ejecutar, examinar/cargar registros, etc.), y visualizadores del contenido hexadecimal de los registros IR y PC y de los biestables indicadores.

- El *modo ampliado* incrementa notablemente las posibilidades del modo real. Se implementa por medio de utilidades y programas que facilitan el acceso directo a cualquier componente de CODE-2: posiciones de memoria, registros, biestables indicadores y puertos de entrada y salida. Entre los programas del modo ampliado, se incluye un módulo de desensamblado, que se comentará más adelante, y que por su propia naturaleza, puede ser considerada como un sub-modo del

modo ampliado, ya que dentro de él cambia el comportamiento de muchos objetos gráficos del entorno ECODE.

En la Figura 1 se incluye una imagen que muestra el aspecto del entorno funcionando en modo ampliado. Se ha tenido especial cuidado, en el diseño de la interfaz de usuario para hacerla lo más didáctica y cómoda posible. La disposición de los componentes gráficos ha sido especialmente pensada para lograr la mayor coherencia posible entre el entorno y la descripción real (física) de CODE-2. También se ha diseñado de forma que el puntero del ratón cambia de aspecto al superponerse a un objeto, indicando este hecho que el usuario puede interactuar directamente con dicho objeto. Además, al descansar unos instantes el puntero del ratón sobre un objeto gráfico relevante, se despliega automáticamente un rótulo indicando su significado.

El entorno está programado en Object Pascal, usando el entorno de Borland Delphi 5.

### 3. Características de ECODE

La Figura 1 muestra la interfaz gráfica de ECODE (versión 1.0.0 para Windows). La imagen se compone de cinco zonas o sub-ventanas, y que de arriba abajo, y de izquierda a derecha son:

**1. Barra de menús**, ubicada en la parte superior de la imagen y contiene las siguientes opciones :

- Archivo**: Desde esta opción es posible abrir un archivo en ensamblador para editarlo, o para traducirlo y guardar el resultado en un archivo o cargarlo directamente en la memoria de CODE-2 para emular su ejecución.
- Entorno**: con esta opción se pueden seleccionar, entre otras utilidades, los distintos modos de funcionamiento del entorno: real, ampliado y desensamblador, así como acceder al convertidor visual de archivos.
- CODE**: esta opción permite interactuar directamente con el emulador de CODE-2.
- Ayuda y manuales**: con esta opción se obtiene información referente al uso de ECODE, al funcionamiento de CODE-2 y un manual de ensamblador de CODE-2. También se incluyen ejemplos de programas y plantillas en ensamblador para facilitar el trabajo del

usuario y la comprensión de los conceptos que se pretenden transmitir al alumno.

- Botones rápidos**: Permiten acceder de forma más ágil, a las rutinas más usadas, estén o no en el menú principal, y son los siguientes: Ver/Ocultar la ampliación del panel frontal, Desensamblar, Variar la velocidad de ejecución, Encender/Apagar CODE-2, Modo paso a paso, Dirección de memoria, Registros, Cargar, Ejecutar, y Continuar.

**2. Sub-ventana de texto**: visualiza el archivo del programa en desarrollo, en lenguaje ensamblado o máquina.

**3. Panel frontal** de CODE-2, que representa al prototipo físico. A esta imagen se le ha añadido en su parte superior tres solapas, para visualizar, alternativamente el panel frontal, los mensajes emitidos por el entorno a lo largo de la sesión, o el contenido de los registros.

**4. Esquema completo** de CODE-2, donde se incluyen, además de los elementos que aparecen en el panel frontal, los contenidos de todos los registros, de la memoria y de los puertos IP1 e IP2 de entrada, y OP1 y OP2 de salida. Todos los contenidos puede modificarse por medio de un editor hexadecimal. Además se permite arrastrar, ampliar y soltar donde el usuario desee la sub-ventana que representa la memoria.

**5. Barra de estado**, informa de la situación actual del programa.

Cuando ECODE funciona en modo real sólo aparece el panel frontal, y cuando funciona en modo ampliado se muestra completamente la imagen de la Figura 1. Por otra parte, el sistema ECODE contiene los siguientes módulos, integrados unos con otros:

- Emulador**: simula el comportamiento de CODE-2.
- Ensamblador**: traduce archivos de ensamblador de CODE-2 a su lenguaje máquina. La salida puede ser presentada en los formatos .HEX (hexadecimal) o .EHC (Easy Hexadecimal Code) descritos en [3].
- Editor hexadecimal**: permite, por ejemplo, cargar directamente un programa en la memoria sin más que situarse con el ratón en la sub-ventana de memoria .
- Desensamblador**: para desensamblar automáticamente un programa cargado previamente en la memoria de CODE-2. El resultado es

simulable posteriormente igual que si hubiera sido el resultado del ensamblaje normal de un programa.

- *Cargador*: Los programas pueden ser almacenados automáticamente en la memoria, después de haber sido traducidos a código máquina por el ensamblador, y de acuerdo con las directivas del programa fuente que especifiquen las posiciones de carga en memoria.
- *Módulo de información*: genera y almacena los mensajes de ECODE, tanto los de error como los de estado (modo de funcionamiento), y se pueden visualizar seleccionando la solapa *Mensajes*.
- *Convertidor Visual de Archivos*: gestiona un cuadro de diálogo con el que el usuario puede realizar la conversión entre los formatos .ASM, .HEX, .EHC, con la excepción de que no podemos pasar de .ASM a cualquier otro.

Algunas de las características del entorno, que enfatizan su fin pedagógico son:

- Visualización de los contenidos de todos los registros y biestables.
- Durante la ejecución de un programa se resalta la posición de memoria donde se encuentra la instrucción en ejecución.
- Interacción directa con los registros y puertos de entrada/salida. Se pueden editar cada uno de estos elementos, seleccionándolos con el ratón.
- Visualización del comportamiento dinámico de todos los contenidos; para lo cual se puede utilizar el modo de ejecución *paso a paso* (CODE-2 entra en modo de espera inmediatamente después de la ejecución de cada instrucción) o regular la velocidad de ejecución utilizando uno de los botones rápidos.

#### 4. Utilización del entorno

Nada más iniciar ECODE, queda listo para empezar a simular. Implícitamente arranca en modo ampliado, y se visualiza la imagen de la Figura 1. Ahora, se puede elegir una de las opciones del menú *Archivo*, si se desea cargar un programa en código máquina, traducir un programa en ensamblador y guardarlo como

código máquina, o editar un programa fuente, traducirlo, depurarlo y cargarlo en la memoria de CODE-2 para su ejecución simulada. Cuando se ejecuta un programa las sub-ventanas de memoria y de texto visualizan la información de forma coherente, resaltando los cambios que se van produciendo en la memoria y en el contador de programa.

En cualquier momento, aún con un programa cargado, podremos editar la memoria, los registros y los puertos, esta "edición en caliente" es muy útil para depurar programas o comprobar de forma sencilla el funcionamiento de cualquier instrucción.

#### 4. Conclusiones

Hemos presentado en esta comunicación las características más notables de ECODE, un entorno integrado de desarrollo (IDE) orientado a la enseñanza, para un Computador Didáctico Elemental (CODE-2). El sistema puede obtenerse en la dirección web:

[http://atc.ugr.es/intro\\_info\\_mcgraw.html](http://atc.ugr.es/intro_info_mcgraw.html)

#### Referencias

- [1] Página web principal del proyecto CODE-2 [http://atc.ugr.es/intro\\_info\\_mcgraw](http://atc.ugr.es/intro_info_mcgraw)
- [2] A. Prieto, A. Lloris, J.C. Torres. *Introducción a la Informática*, 3ª Edc. Cap 6-7, McGraw-Hill 2002.
- [3] A.Prieto, F.J.Pelayo, F.Gómez, J.Ortega, A.Cañas, A.Martínez, F.J.Fernández. *Un computador didáctico elemental*. Actas de la VIII Jornadas de Enseñanza Universitaria de la Informática (JENUI'2002). Cáceres 10-12 Julio 2002. (en este mismo libro)
- [4] A. Martínez. *Diseño de un Entorno Integrado de Desarrollo con emulador y compilador cruzado para el computador CODE-2*, Proyecto fin de carrera de Ingeniero Electrónico, Dpto. Arquitectura y Tecnología de Computadores, Universidad de Granada. Octubre 2001.
- [5] *IEEE Standard for Microprocessor Assembly Language*. IEEE Std. 694-1985. The Institute of Electrical and Electronics Engineers, Inc. New York., June 30, 1985.