

INCORPORACIÓN DE BDD A LA DOCENCIA DE CIRCUITOS DIGITALES

C. Baena¹, M.P. Parra¹, J.J. Baena y J.M. Quintana²

1 Departamento de Tecnología Electrónica

2 Departamento de Electrónica y Electromagnetismo

Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla

Tfno: 95-4552785; Fax: 95-4552764; e-mail: baena@cnm.us.es

1 y 2 también Dpto. Diseño Analógico, (IMSE-CNM)

RESUMEN.- En este trabajo se propone la inclusión de una nueva herramienta de descripción de funciones de conmutación, los BDD, en cursos sobre circuitos digitales. Los BDD presentan un gran auge en los últimos años en materias de investigación por lo que, previsiblemente se implantarán muy pronto en enseñanzas universitarias. Para ello, se ha desarrollado un entorno informático para el manejo de BDD, con una presentación amigable al usuario.

1.- INTRODUCCIÓN

El estudio de los sistemas digitales obliga a conocer y utilizar adecuadamente formas de descripción de funciones de conmutación. Existen muchas herramientas que realizan esta descripción de manera más o menos eficiente. Entre las más clásicas pueden citarse las expresiones algebraicas y las formas tabulares (tablas de verdad, mapas de Karnaugh o binarios, etc). La enseñanza de estas formas clásicas suelen estar incluidas en los cursos de Fundamentos de Computadores, Diseño Lógico, Circuitos y Sistemas Digitales y equivalentes. Sin embargo, estas descripciones de funciones de conmutación crecen en complejidad muy rápidamente (conforme aumenta el número de variables) con lo que "explotan" relativamente pronto.

Recientemente han surgido diversas alternativas a esas descripciones de las que, en este trabajo, nos vamos a centrar en los Diagramas de Decisión Binaria (BDD)[1,2]. Esta forma de descripción de funciones de conmutación ha mostrado su eficiencia en campos muy distintos del diseño VLSI ya que sólo presenta un crecimiento lineal con respecto al número de entradas. Los BDD se han usado en problemas de optimización y, en general, en el contexto de la síntesis lógica. Por otra parte, es una descripción suficientemente simple como para incluirla en un curso universitario. Nuestro objetivo en este trabajo es proponer una forma de cómo realizarlo.

Esta comunicación se estructura en cuatro apartados. Tras este apartado de introducción, se resumen las principales características de la representación de funciones lógicas mediante BDD y se presenta una propuesta para su incorporación a la docencia. En el tercer apartado se aborda la descripción de una herramienta informática que se ha desarrollado para la representación y manipulación de BDD y se la relaciona con nuestra propuesta docente. Por último se resumen las conclusiones.

2.- DIAGRAMAS DE DECISIÓN BINARIA

Un BDD es un grafo con estructura de árbol donde podemos distinguir dos tipos de nodos. Los nodos internos, que están directamente relacionados con cada una de las variables de la función, y los nodos terminales, que representan a los valores lógicos (0 y 1) que puede tomar la función. De cada nodo interno parten dos arcos, cada uno de los cuales representa uno de los dos valores posibles de la variable a la que representa. El arco etiquetado con T (then) indica que la variable correspondiente toma el valor lógico 1 y el etiquetado con E (else) indica que toma el valor lógico 0. En la Figura 1 se muestra un BDD como ejemplo. En él se representa a la función de dos variables $f(a,b) = a \cdot b$. Cada uno de los nodos actúa como una llave if-then-else cuya variable asociada, a ó b, puede conmutar entre su arco T y su arco E. Así, podemos ver en el gráfico que para llegar al nodo terminal 1 desde el nodo superior es necesario pasar por el arco T de los nodos a y b. Por tanto, $f(a,b) = 1$ si $a = b = 1$ y $f(a,b) = 0$ en los demás casos.

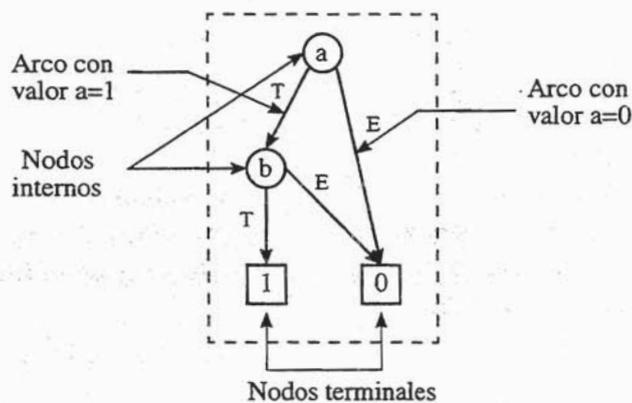


Figura 1.- Ejemplo de un BDD simple: $f(a,b) = a \cdot b$

Este tipo de diagrama aporta una visión diferente de la que se ha presentado tradicionalmente, ya que se trata de una representación gráfica frente a las basadas en mapas o ecuaciones. Además, debemos destacar su simplicidad conceptual, lo que permite que el alumno se familiarice sin muchas dificultades con ella.

De forma similar a como se ha presentado previamente el BDD de la Figura 1 se puede introducir este concepto en un curso de Circuitos Digitales. Un momento adecuado puede ser cuando se aborda la representación de funciones lógicas a través de otros métodos más convencionales: tablas de verdad, mapas... También se pueden mostrar los BDD de los operadores lógicos básicos (or, and, nor, nand, etc.) así como los de diversas funciones simples. En primera instancia, el análisis de estos BDD simples serviría como una primera introducción y permitiría una comparación desde el punto de vista cualitativo con las otras formas de representación.

En nuestra propuesta también se retomarían los BDD al presentar conceptos clásicos como la expansión de funciones mediante el teorema de Shannon, materia ésta contenida en un curso estándar. Según dicho teorema, cualquier función $f(x_1, x_2, \dots, x_i, \dots, x_n)$ puede expresarse como:

$$f(x_1, x_2, \dots, x_i, \dots, x_n) = \bar{x}_i f(x_1, x_2, \dots, 0, \dots, x_n) + x_i f(x_1, x_2, \dots, 1, \dots, x_n)$$

donde $f(x_1, x_2, \dots, 0, \dots, x_n)$ y $f(x_1, x_2, \dots, 1, \dots, x_n)$ se denominan residuos.

El concepto de BDD puede relacionarse con dicha expresión, ya que el papel jugado por cada uno de los nodos sería el de la variable x_i representada en ellos y cada residuo de la función se

correspondería con cada arco. En la Figura 2-a se muestran estas correspondencias. Como consecuencia de lo anterior, la representación de una función a través de un BDD está también directamente relacionada con su implementación mediante una red de multiplexores de una entrada de control. En la Figura 2-b se muestra este paralelismo para el caso de la función $f(a,b) = a \cdot b$.

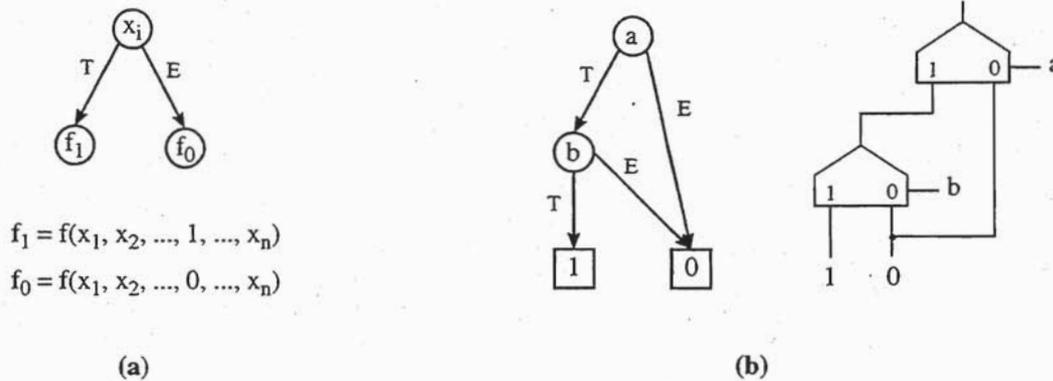


Figura 2.- (a) Teorema de expansión mediante BDD. (b) Relación entre el BDD y la implementación mediante MUX2:1 para la función $f(a, b) = a \cdot b$

Posteriormente, durante el curso y en un tema específico se presentarían las principales características de la descripción de funciones mediante BDD, pasando así de la descripción puramente cualitativa a una más detallada.

A continuación, mencionaremos los principales conceptos a desarrollar. Estos serán fundamentalmente los que tienen que ver con las operaciones de complementación, reducción y ordenación de variables.

Dado el BDD de una función, el correspondiente a la función complementaria se puede obtener muy fácilmente sin más que intercambiar los nodos terminales. No obstante, se debe presentar lo que se denomina atributo complemento, mostrando que con su uso basta con marcar el nodo superior del BDD para indicar que la función se ha complementado. Si esta marca aparece en arcos internos del árbol, indica que los subárboles que parten de ellos están complementados. Con el atributo complemento es posible además eliminar uno de los nodos terminales, que no sería más que el complemento del otro.

En cuanto a la reducción de BDD ha de mostrarse la técnica de localización de subgrafos con idéntica forma dentro de un árbol. Dos subgrafos de este tipo pueden sustituirse por uno solo. Tras una reducción así pueden aparecer nodos redundantes, es decir, nodos cuyos arcos T y E conducen al mismo lugar con lo que pueden eliminarse.

Tras analizar algunos BDD para su reducción se pasa a discutir la importancia de la ordenación de variables elegida a la hora de representar una función mediante un BDD. Dicho orden influye sobre el tamaño de éste. Se mostrarán algunos ejemplos significativos.

La siguiente idea a desarrollar es que teniendo en cuenta el atributo complemento y aplicando sucesivamente las operaciones de reducción se llega a obtener un BDD de tamaño mínimo. Se puede garantizar que dicho BDD es único para una ordenación de variables dada. Estos resultados no se demuestran formalmente. Simplemente se enuncian para enfatizar que, gracias a que para una función dada existe un único BDD (canonicidad), las operaciones de chequeo de equivalencia entre funciones se facilitan. Otra operación que resulta muy simple es la de encontrar los unos de la función. Este problema es equivalente a recorrer el grafo desde el nodo raíz al nodo terminal por algún camino que incluya un número par de atributos complementos y esto se mostraría con diferentes ejemplos.

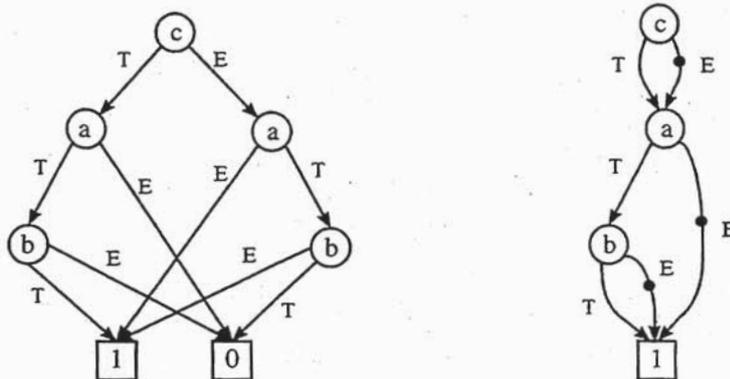


Figura 3.- Ejemplo de reducción de un BDD: $f(a,b,c) = a \cdot b \cdot c + \bar{a} \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c}$

3.- PROGRAMA INFORMÁTICO PARA EL TRATAMIENTO DE LOS BDD

La obtención de BDD reducidos no se realiza de forma manual. Existen varios paquetes software que son ampliamente usados para este fin. Estos han sido desarrollados por diversos grupos de distintas universidades [3,4]. En general, el BDD obtenido se ofrece en algún formato de texto que depende del paquete utilizado. En la Figura 4 se muestran dos ejemplos de dichos formatos que describen el BDD de la función $f(a,b,c) = a \cdot b \cdot c + \bar{a} \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c}$ (Figura 3). Estos han sido obtenidos a través de los dos paquetes software referenciados anteriormente.

```

index 0 is v#0
index 1 is v#1
index 2 is v#2
ID = 0x10c6  index = 0  T = 0x10bf  E = !0x10bf
ID = 0x10bf  index = 1  T = 0x10be  E = 0
ID = 0x10be  index = 2  T = 1      E = 0

```

```

.model DD
.inputs 0 1 2
.outputs f0
.names 15c91
1
.names 2 15c91 15c91 15c97
11- 1
0-0 1
.names 1 15c97 15c91 15c98
11- 1
0-0 1
.names 0 15c98 15c98 15c99
11- 1
0-0 1
.names 15c99 f0
1 1
.end

```

Figura 4.- Ejemplo de formatos de texto para BDD

Construir el grafo a partir de un formato de texto es una tarea tediosa incluso para árboles pequeños (~10 nodos). Por tanto, para un número medio de variables no es posible realizar esta tarea de forma manual, es por ello que como complemento a este tratamiento teórico hemos creído interesante desarrollar una herramienta software de ayuda mediante un proyecto fin de carrera. Consiste en un entorno informático que permite obtener en pantalla y de forma gráfica el BDD de una función dada, admitiendo como entrada diversos formatos para dicha función como, por ejemplo, tablas de verdad, ecuaciones algebraicas, etc. El propio entorno se encarga de traducir estos formatos adecuadamente de manera que puedan ser procesados por

el paquete concreto de funciones que esté utilizando.

En particular, se ha realizado la herramienta de manera que sea compatible con dos paquetes de funciones [3,4]. Las funciones están escritas en lenguaje C, y los formatos de texto que proporcionan son distintos (Figura 4). Nuestro entorno interpreta esta salida de texto y representa gráficamente el correspondiente árbol. En la Figura 5 se muestra un diagrama de bloques de la herramienta. En ella podemos distinguir dos partes bien diferenciadas: *generador* e *interfaz gráfico*.

La primera toma como entrada un fichero que contiene la representación de una o varias funciones lógicas. Las entradas que se permiten en el entorno son las escritas en formato *eqntott* (expresiones algebraicas) y *espresso* (tablas de verdad, suma de productos o producto de sumas reducidas). A partir de ellos el módulo *generador* produce un fichero de código fuente en lenguaje C que describe los BDD haciendo uso de las funciones de librería de los paquetes mencionados. El usuario podrá seleccionar la librería que prefiera. Posteriormente basta compilar ese código y se obtendrá el texto que describe al BDD.

El segundo módulo del entorno (*interfaz gráfico*) interpreta la salida del módulo anterior y representa el dibujo del BDD. Además podemos conservar el fichero de texto producido en el primer módulo. Esto permite el acceso directo al segundo módulo de la herramienta.

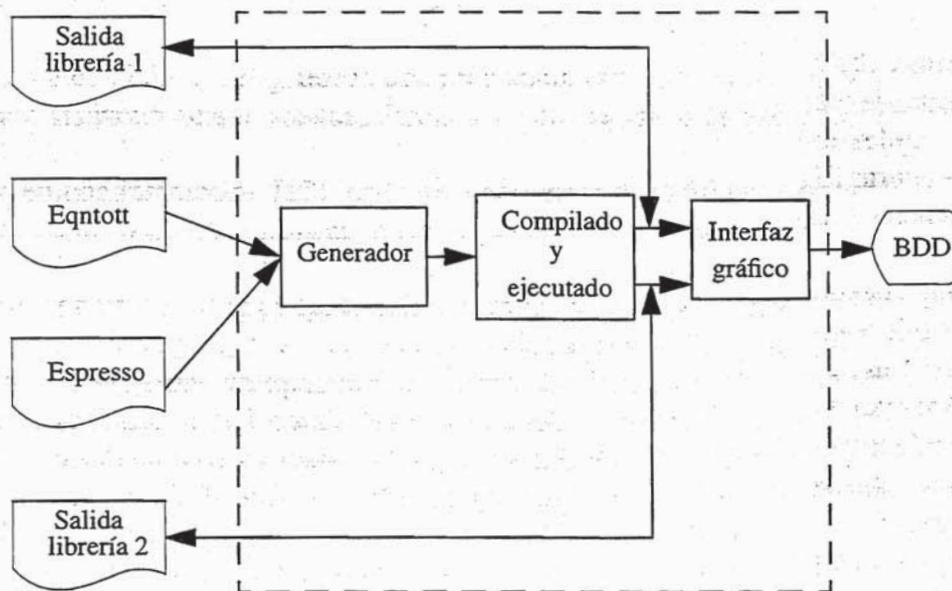


Figura 5.- Diagrama de bloques del entorno gráfico

El *interfaz gráfico* posibilita realizar diversas operaciones sobre el árbol o árboles obtenidos. A modo de ejemplo, podemos citar la posibilidad de hacer modificaciones de un árbol cambiando el orden en que aparecen cada una de las variables. También, se ofrece un modo gráfico de localizar cubos (productos que implican a la función) en las funciones representadas. Esto puede realizarse de modo independiente o de modo múltiple. Es decir, podemos obtener los cubos de una única función, o encontrar cubos comunes a funciones distintas. Otra capacidad del entorno es que permite la realización de operaciones lógicas entre árboles y muestra el resultado. Además es posible realizar operaciones de zoom y escalado sobre los gráficos. En la Figura 6 se representa un ejemplo de ventana del *interfaz gráfico* en la que pueden apreciarse dos zonas diferentes. En una de ellas aparece el BDD de la función que se esté representando. En la otra zona se controlan algunas operaciones como por ejemplo las de escalado o de ordenación de variables, así como características de color de los nodos y arcos. También

en esta sección de pantalla se ofrece información del número de BDD cargados en ese momento, así como el número de nodos, minterminos y caminos del grafo.

El entorno que presentamos es modificable ante nuevas versiones del paquete de funciones sobre el que se apoya ya que se ha realizado de la forma más estructurada posible.

Con esta herramienta se facilita el estudio y manipulación de BDD, por tanto, nuestra propuesta es utilizarla para profundizar en el estudio de los mismos a nivel práctico. Además podría utilizarse como herramienta de autoaprendizaje ya que se ha puesto énfasis en dotarle de una presentación amigable al usuario y ofrece la posibilidad de consultar una guía de ayuda on-line.

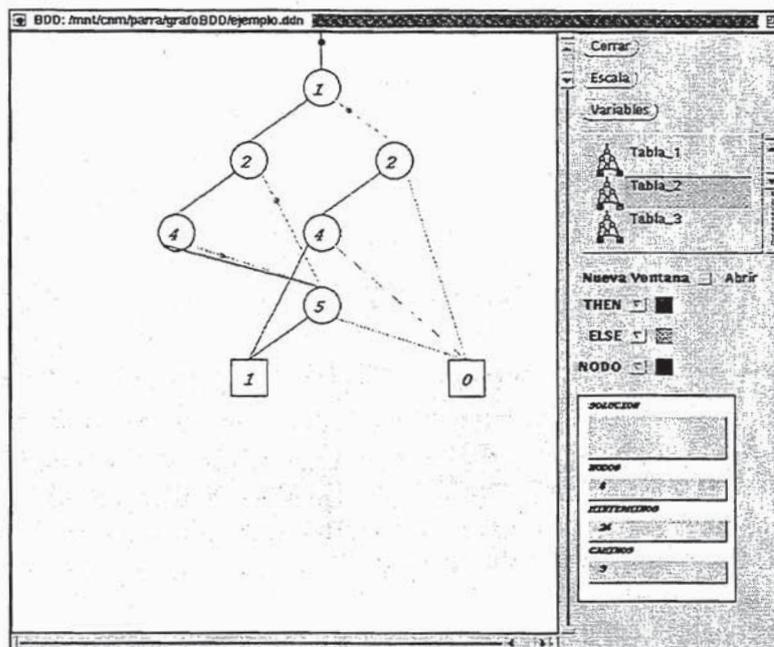


Figura 6.- Ventana de presentación del interfaz gráfico

4.- CONCLUSIONES

En esta comunicación, hemos realizado una propuesta para la incorporación de los diagramas de decisión binaria (BDD) a la docencia de asignaturas relacionadas con el diseño lógico. Hemos enumerado contenidos concretos que pueden ser desarrollados en dichas asignaturas. Asimismo, se ha presentado una herramienta informática amigable al usuario que facilita el tratamiento y el estudio de esta forma de representación de funciones.

5.- REFERENCIAS

- [1] Bryant, R.E. "Graph-Based Algorithms for Boolean Function Manipulation". *IEEE Transactions on Computers*, Vol. c-35, n^o 8, pp. 677-691. 1986.
- [2] Akers, S.B. "Binary Decision Diagrams". *IEEE Transactions on Computers*, Vol. c-27, pp 509-516. 1978.
- [3] Shiple, T. "Binary Decision Diagram (BDD) Package, Version 2.3". University of California, Berkeley. 1992.
- [4] Somenzi, F. "CUDD: CU Decision Diagram Package. Release 1.0.4". University of Colorado, Boulder. 1995.