

Uso de Herramientas CAD Aplicadas al Diseño de Circuitos Digitales Combinacionales

Use of CAD Tools Applied to the Design of Combinational Digital Circuits

CÉSAR AUGUSTO ROMERO MOLANO*
UNIVERSIDAD DE LOS LLANOS

Recibido: 9 de junio de 2015/Aceptado: 26 de agosto de 2015

RESUMEN

Este trabajo discute la necesidad de incluir el uso de herramientas software como estrategia pedagógica en el desarrollo temático del curso circuitos digitales, con el objeto que el estudiante pueda diseñar aplicaciones del tipo combinacional con un gran número de variables de entrada (seis o más variables) en un tiempo muy corto y con resultados que puedan ser implementados en circuitos integrados o en dispositivos lógicos programable.

Palabras clave: Algebra Boole, Diseño de circuitos digitales, Enseñanza.

ABSTRACT

This paper discusses the need of include the software tools as a pedagogic strategy in the thematic development of the digital circuits subject, with the objective the student will be able to design combinational type applications with a big quantity of variables (six or more) in a short time and with results than can be implements with integrated circuits or in programmable logic devices.

Keywords: Boole algebra, Design of Digital circuits, Teaching.

* Docente de planta Universidad de los Llanos del programa de Ingeniería Electrónica, Ingeniero Electrónico, Especialista en Redes de datos. cesar.romero@unillanos.edu.co

1. INTRODUCCIÓN

El diseño de circuitos digitales es uno de los primeros cursos teórico-práctico al cual se enfrenta un estudiante de ingeniería electrónica en sus primeros años de formación profesional. El mismo plantea como reto aplicar conceptos matemáticos como es el álgebra de Boole. El siguiente reto para un estudiante es aplicar los teoremas y propiedades de esta, en la simplificación de ecuaciones lógicas enfrentándose con el inconveniente de la no existencia de un procedimiento que le permita llegar a una óptima simplificación, el obtener esta, es producto del manejo y el aplicar el álgebra.

La primera alternativa de solución a este problema es el método del mapa, el cual es usado como procedimiento de representación para funciones de Boole dadas en términos mínimos o máximos, las cuales se pueden simplificar directamente. Esta técnica fue propuesta primero por Veitch (1952) y modificada ligeramente por Karnaugh (1953).

La segunda alternativa de solución es la utilización del método llamado del tabulado, formulado primero por Quine (1952) y más tarde mejorado por Mc-

Cluskey (1956), este método plantea un procedimiento lógico estructurado en pasos, el cual una vez aplicado de forma rigurosa entregará una ecuación de Boole simplificada.

Las diferentes alternativas de simplificación nombradas anteriormente (álgebra de Boole, mapa y tabulado) son la alternativa tradicional para el estudio y diseño de circuitos combinacionales. Las mismas son muy efectivas para el diseño de estos con un bajo número de variables de entrada (menor a cinco variables), pero cuando se requiere diseñar este circuito con más de seis variables de entrada, se requiere un gran dominio de estas técnicas de simplificación.

Lo que propone el presente artículo es aplicar herramientas software al diseño e implementación de circuitos combinacionales lo cual plantea un diseño que no requieren un gran dominio de las técnicas de simplificación tradicionales.

2. DISEÑO DE CIRCUITOS COMBINACIONALES

Estudiado desde un enfoque tradicional y, aplicando álgebra de Boole, básicamente se logra agotando las siguientes cuatro fases, las cuales son: Tabla de

verdad del problema planteado (toda situación que implique un problema combinacional se puede llevar a una tabla de verdad), obtención de la ecuación Booleana solución al problema, simplificación de la ecuación anterior.

Este mismo problema de diseño aplicando herramientas de software se logra en las siguientes tres fases las cuales son: tabla de verdad del problema planteado, utilización de software como Multisim de national Instruments o ISE design tools de Xilinx y finalmente se obtiene el circuito combinacional solución, el cual se podrá implementar con circuitos integrados tradicionales TTL, CMOS o podrá también ser realizado en dispositivos lógicos reconfigurables como CPLD o FPGA.

Para continuar el desarrollo del tema se planteará el siguiente ejemplo de diseño.

Diseñar un circuito combinacional que genere el bit de paridad par para un dato binario de cuatro bit de entrada.

2.1. Diseño del circuito combinacional propuesto enfoque tradicional

Como se mencionó anteriormente este tipo de diseño se logra agotando las siguientes cuatro fases.

Primera Fase: Tabla de verdad

Por ser el problema planteado un circuito combinacional de cuatro variables de entrada se tendrá diez y seis combinaciones de entrada posible y una única salida. Esta tabla de verdad se ilustra a continuación Tabla 1.

Tabla 1. Tabla de verdad

A	B	C	D	Fout
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Segunda Fase: Ecuación de Boole

De la Tabla 1 se obtiene la ecuación de Boole para Fout por sumatoria de términos mínimos así:

$$Fout(A, B, C, D) = \sum (1,2,4,7,8,11,13,14)$$

Tercera Fase: Simplificación

La simplificación de Fout se puede obtener

ner mediante la aplicación del álgebra de Boole o mediante el estudio de los mapas de Karnaugh. Estas dos alternativas se desarrollaran a continuación.

La simplificación de Fout usando álgebra de Boole es:

$$Fout(A,B,C,D) = A \oplus B \oplus C \oplus D$$

El mapa de Karnaugh para Fout es:

	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

En el anterior mapa se puede observar que no es posible realizar asociaciones de unos o ceros, luego esta técnica de simplificación no es la más adecuada para obtener una ecuación simplificada para Fout.

$$Fout(A,B,C,D) = (\bar{A}\bar{B}\bar{C}D) + (\bar{A}\bar{B}C\bar{D}) + (\bar{A}B\bar{C}\bar{D}) + (\bar{A}B.C.D) + (A.B.\bar{C}.D) + (A.B.C.\bar{D}) + (A.\bar{B}.\bar{C}.\bar{D}) + (A.\bar{B}.C.D)$$

Cuarta Fase: Circuito combinacional

Con los resultados obtenidos en la fase tres se puede concluir que el circuito que presenta un menor uso de compuestas es

el hallado por simplificación usando álgebra de Boole ya que solo utilizaría una compuerta X-OR de cuatro entradas.

2.2. Diseño del circuito combinacional propuesto, usando herramientas de software

Como se mencionó anteriormente este tipo de diseño se logra agotando tres fases.

Primera Fase: Tabla de verdad

La tabla de verdad para el problema planteado no cambia y es la que se observa en la Tabla 1, la cual tiene cuatro entradas y una salida.

Segunda Fase: Uso de herramientas software

Una vez se tiene identificada la tabla de verdad el siguiente paso es hacer uso de la herramienta PLD design disponible en multisim, este instrumento permite elaborar un proyecto el cual se puede implementar en dispositivos lógicos programables, para el uso de esta se sugiere ver la ayuda del software. Es necesario aclarar que para su uso, no es indispensable para desarrollar la aplicación-solución al problema planteado se puede hacer uso de la herramienta logic converter igualmente disponible en mul-

tisim esta herramienta permite ingresar la tabla de verdad del problema que pretendemos solucionar y la herramienta entrega la ecuación del problema escrita en términos mínimos o la ecuación simplificada y el circuito combinacional solución al problema usando compuertas AND y compuertas OR.

La Figura 1 muestra la tabla de verdad y la ecuación simplificada representada en productos de sumas proporcionadas por la herramienta logic converter.

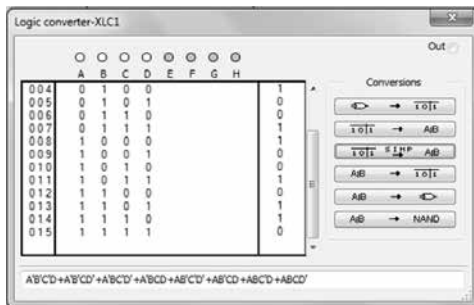


Figura 1. Imagen Logic Converter

Como se puede observar el procedimiento de simplificación se redujo a solo usar la herramienta la cual hace todo el trabajo de simplificación.

Tercera Fase: Circuito combinacional

Con la ayuda de la herramienta logic converter se puede obtener el circuito solución que puede ser implementado

utilizando circuitos integrados TTL o CMOS, basta con realizar un clic en el botón correspondiente de la herramienta y obtenemos el circuito que se observa en la Figura 2.

Para llevar esta misma solución a un dispositivo lógico programable lo que se debe hacer es utilizar la opción export to PLD del menú transfer la cual entrega dos archivos con extensión VHD, los cuales se pueden implementar en el dispositivo lógico programable que se desea usando un software para tal fin como lo es ISE design tools de xilinx.

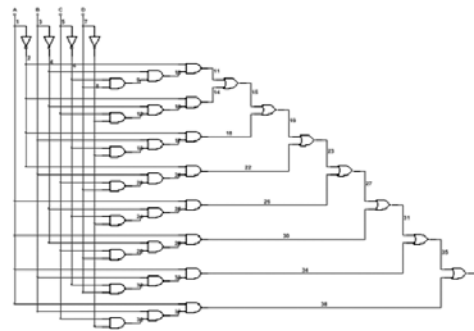


Figura 2. Circuito solución

Como una segunda alternativa de solución a la fase dos, llamada en este artículo uso de herramientas de software pero esta vez pensando llevar la solución a un dispositivo lógico programable como un CPLD o FPGA, se puede hacer uso del

software ISE design tools de xilinx, donde se procede a crear un proyecto el cual estará formado por un módulo VHDL que implementara la tabla de verdad del problema combinacional que se está solucionando ver Tabla 1, el código VHDL solución para el problema planteado en este artículo se puede observar en la Figura 3, en el mismo se observa que se definió un puerto de entrada A como un vector de cuatro posiciones donde el bit LSB de este vector hace referencia a la variable D de la tabla de verdad mostrada en la Tabla 1.

Una vez escrito el código mostrado en la Figura 3 se procede a realizar la verificación de la sintaxis del mismo por medio de la herramienta Synthesize-XST del software ISE, una vez realizado este proceso de verificación se pasa a observar el contenido del módulo solución utilizando la herramienta view Technology schematic Figura 4.

En el mismo se observa el bloque de la tabla “lookup table” que da solución al problema planteado. Realizando doble clic sobre el módulo lut se obtiene una ventana la cual muestra el esquemático, ecuación, tabla de verdad y mapa de Karnaugh solución.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity modulo is
  Port ( A : in  STD_LOGIC_VECTOR(3 DOWNTO 0);
        Fout : out  STD_LOGIC);
end modulo;

architecture Behavioral of modulo is
begin
  PROCESS (A)
  begin
    case A IS
      when "0000"=>Fout<='0';
      when "0001"=>Fout<='1';
      when "0010"=>Fout<='1';
      when "0011"=>Fout<='0';
      when "0100"=>Fout<='1';
      when "0101"=>Fout<='0';
      when "0110"=>Fout<='0';
      when "0111"=>Fout<='1';
      when "1000"=>Fout<='1';
      when "1001"=>Fout<='0';
      when "1010"=>Fout<='0';
      when "1011"=>Fout<='1';
      when "1100"=>Fout<='0';
      when "1101"=>Fout<='1';
      when "1110"=>Fout<='1';
      when others=>Fout<='0';
    end case;
  end process;
end Behavioral;

```

Figura 3. Código VHDL circuito solución

Para finalmente llevar esta solución a un dispositivo lógico programable como un CPLD o FPGA basta con terminar los procesos de Implement design, Generate programming file y configure Target Device del software ISE, para mayor detalle en el uso del software ISE Project navigator ver la ayuda dispuesta en el mismo.

El autor deja a disposición del público en general una serie de videos tutoriales de las herramientas nombradas en el artículo las cuales se encuentran alojados en youtube bajo política de acceso y uso libre, la dirección del canal del autor

es https://www.youtube.com/user/ceromal175/featured?view_as=public

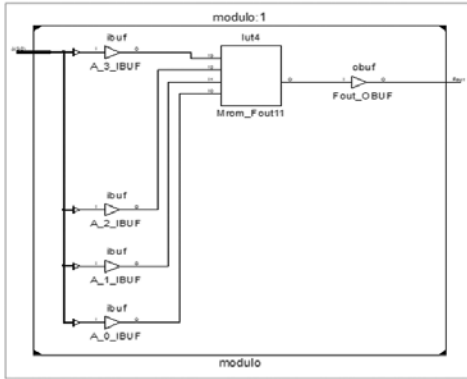


Figura 4. Módulo solución

3. CONCLUSIÓN

Con este artículo se busca mostrar a la comunidad académica, entendida como profesores y estudiantes de ingeniería electrónica o de sistemas, que es posible abordar el diseño de circuitos digitales desde el uso de herramientas software las cuales optimizan el proceso de diseño.

El uso de software como Multisim de national instruments o ISE Design suite de Xilinx, facilitan el proceso de diseño para circuitos digitales combinacionales o secuenciales y generan un nuevo escenario en el aula de clase ya que el profesor puede dejar el largo y tedioso proceso llamado optimización de circuitos a este tipo de herramientas y centrar

los temas de enseñanza y discusión del salón de clase a estudios de casos y al análisis de los resultados entregados por estas herramientas.

4. REFERENCIAS BIBLIOGRÁFICAS

- Karnaugh, M. (1953). A map method for synthesis of combinational logic circuits. *Trans. AIEE, Comm. and Electronics*, 72, Parte I, 593-599.
- McCluskey, E. J. (1956). Minimization of Boolean functions. *Bell system tech. J.*, 35(6), 1417-1444.
- Quine, W. V. (1952). The problem of simplifying truth functions. *Am. Math. Monthly*, 59(8), 521-531.
- Veitch, E. W. (1952). A chart method for simplifying truth functions. *Proc. of the ACM*, 127-133.