

Luis Alejandro Díaz Bayona.

Licenciado en Educación mención Matemática, Magíster en Educación Matemática. Docente ordinario adscrito a la Cátedra de Cálculo del Departamento de Matemática y Física. Facultad de Ciencias de la Educación, Universidad de Carabobo (UC).

Correo: prof Judi@gmail.com

Teléfono: +584128543719

**UNA APLICACIÓN PRÁCTICA DE LA LÓGICA PROPOSICIONAL Y EL ÁLGEBRA DE BOOLE
A CIRCUITOS SERIE-PARALELO**

Resumen:

La filosofía analítica se fundamenta esencialmente en las ciencias naturales, la lógica matemática y el análisis del lenguaje, es decir, el análisis lógico del lenguaje científico o el análisis lingüístico del lenguaje común; con el propósito de esclarecer los conceptos filosóficos y científicos. En este orden de ideas el propósito del presente artículo es fundamentar desde la filosofía analítica una aplicación práctica de la lógica proposicional y el álgebra booleana en los circuitos serie-paralelo que se estudian en física especialmente en los circuitos eléctricos.

Palabras Clave: Polinomios lógicos, circuitos series-paralelo, lógica proposicional, filosofía analítica.

Introducción:

La Matemática es arte y ciencia. Es un lenguaje universal que es necesario hablar para podernos explicar la naturaleza. Se diferencia de las ciencias fácticas en que mientras en aquellas se descubre y se estudia, en la Matemática se crea. El "objeto matemático" existe cuando el que "hace matemática" lo ha creado. La Matemática, entonces, se hace ciencia cuando el creador del objeto matemático se dedica a estudiarlo. En consecuencia, por naturaleza, la Matemática es una ciencia libre y por su diversidad de aplicación, la más útil de todas.

Siguiendo esta línea de discurso, interesa hablar de George Boole (1815–1864) quien fue el primero en estudiar cómo esquematizar las operaciones lógicas mediante una estructura algebraica. En 1847 publicó el libro *A Investigation of the Laws of Thought* ("Una investigación de las leyes del pensamiento") donde utilizó reticulados para formalizar la lógica y el cálculo de proposiciones. A Boole le preocupaba más la forma que el contenido, por lo que se interesó más en cómo hacía los cálculos que en qué era lo que calculaba. Augustus D’Morgan (1806–1871) continuó el trabajo de Boole. D’Morgan observó que el Principio de Dualidad se cumplía en la

Teoría de Conjuntos, como lo ilustró con las llamadas Leyes de D’Morgan. Él, al igual que Boole, consideraba que la Matemática era el estudio de símbolos y operaciones abstractas.

La Lógica, a la par que la Teoría de Conjuntos, recibieron el aporte de matemáticos tales como Alfred North Whitehead (1861–1947), Bertrand Russell (1872–1970), y David Hilbert (1862–1943). En el libro *Principia Mathematica* (“Los principios de la Matemática”), Whitehead y Russell se propusieron mostrar la conexión que consideraban existía entre las matemáticas y la lógica, valiéndose de la deducción del sistema de números naturales a partir de las reglas de la lógica formal. Planteamiento: Si es posible determinar los números naturales a partir de la lógica misma, entonces también puede hacerse con gran parte de las matemáticas restantes. Lamentablemente para ambos, Kurt Gödel (1906–1978), demostró que siempre existirán problemas “indecidibles” en cualquier sistema axiomático. Lo que significaba que en cualquier sistema matemático importante, siempre se hallarán enunciados que no puedan ser demostrados ni refutados. Esta investigación, básica en matemáticas puras, con el tiempo se volvió indispensable para una gran diversidad de aplicaciones.

La filosofía analítica, postulada por diversos filósofos como Russell, Whitehead, Wittgenstein, Moore, entre otros, esta disciplina o tradición filosófica también es conocida como “Atomismo Lógico” representado por las conferencias de Russell de 1918, recogidas como “La filosofía del atomismo lógico” (1918) en *Lógica y Conocimiento* (1956) y por el *Tractatus Lógico-philosophicus* de L. Wittgenstein (1921). También esta tradición recibió influencias muy marcadas de la filosofía del llamado “Círculo de Viena” la cual se constituyó en torno a la figura de Moritz Schlick, catedrático de filosofía de la Universidad de Viena (de allí su nombre). Pero en realidad no hay un consenso exacto de quienes son los autores, ya que como se vio es un conglomerado de autores que tienen como aporte afirmar que el lenguaje está íntimamente relacionada con la filosofía, todos los problemas deben ser resueltos lingüísticamente, haciendo énfasis en la pregunta o cuestión a resolver para evitar los pseudo problemas. También afirma que se debe analizar los conceptos para tener una mejor comprensión de los hechos a estudiar y su significado en el lenguaje.

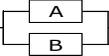
Desde este punto de vista, en el presente artículo, se hace una fusión de dos objetos matemáticos (polinomios lógicos) y (circuitos serie-paralelo) a través de un “lenguaje” común, la lógica proposicional, ya que este lenguaje permite manipular, operar, desglosar elementos abstractos como variables, conectivos y sus valores veritativos o valores lógicos a través de un sistema numérico (números binarios: 0 y 1) y llevarlo a una aplicación física o de mundo real como los circuitos eléctricos, entendidos como asociaciones de dispositivos serie-paralelo y por supuesto hacer el proceso inverso. Es por ello que el Álgebra Booleana y la Lógica se volvieron parte fundamental en el diseño de circuitos integrados a gran escala, los cuales hoy en día se encuentran en los circuitos de cualquier dispositivo electrónico.

Circuitos Serie-Paralelo y su conversión a Polinomios Lógicos:

En sí, el Álgebra de Boole es un método para simplificar los circuitos lógicos cuando se trabaja en electrónica digital. En la lógica booleana, al igual que en la lógica matemática donde una variable proposicional, como variable discreta, solo puede tener dos valores: Verdadera (V) o Falsa (F), los estados de los *dispositivos* (variables para el caso) solo pueden ser dos: Encendido (1) y Apagado (0).

Las asociaciones de *dispositivos en serie* se pueden traducir a *conjunciones*, es decir un circuito dibujado como:  se traduce como $A \wedge B$, ya que la única manera para que haya transmisión de corriente es que los dispositivos A y B estén encendidos, y la tabla lógica se adapta con total naturalidad como se evidencia a continuación:

A	\wedge	B
1	1	1
1	0	0
0	0	1
0	0	0

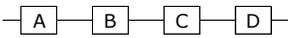
Las asociaciones de dispositivos en paralelo se pueden traducir a *disyunciones*, es decir un circuito dibujado como:  se traduce como $A \vee B$, ya que la única manera para que no haya transmisión o flujo de corriente es que los dispositivos A y B estén apagados, y la tabla lógica se adapta con total naturalidad como se evidencia a continuación:

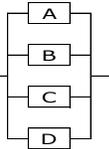
A	\vee	B
1	1	1
1	1	0
0	1	1
0	0	0

Ahora debemos definir un tercer elemento que corresponde al estado contrario de un dispositivo, en lógica a eso se la llama negación (por ejemplo, A' es la negación de A) y su tabla lógica sería la siguiente:

A	A'
1	0
0	1

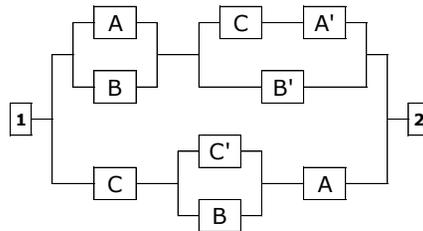
Con estos sencillos principios es totalmente válido convertir o transformar un circuito eléctrico en un modelo matemático (circuito lógico) que permita su estudio usando leyes del Álgebra de Boole, convirtiendo dichos circuitos en polinomios lógicos. Se debe tomar en cuenta que las conexiones en serie se agrupan de dos en dos y de izquierda a derecha, y las conexiones en paralelo se agrupan de dos en dos de arriba hacia abajo. Esta norma permite trabajar con agrupaciones que posee varios dispositivos y se usarían diversos signos de agrupación (paréntesis, corchetes, llaves, barras), entre otros, para darle un orden a la hora de elaborar la tabla de dicho circuito.

Por ejemplo la traducción de la siguiente serie  sería: $[(A \wedge B) \wedge C] \wedge D$, ya que se está agrupando de dos en dos dentro de cada signo de agrupación y la agrupación se hace de izquierda a derecha.

En cambio, para el siguiente paralelo  sería: $[(A \vee B) \vee C] \vee D$, ya que se agrupa de dos en dos y de arriba hacia abajo.

La única manera de apreciar los alcances de esta propuesta sería a través de configuraciones más elaboradas, donde se aprecien combinaciones interesantes de las asociaciones serie-paralelo pero los principios explicados del orden de agrupación se debe respetar y cumplir a cabalidad; si no se hace de esa manera, no se obtendrían conclusiones válidas. De momento se abordará la construcción del polinomio lógico de un circuito eléctrico sencillo y luego se añadirá el concepto de "árbol de programación" el cual es una guía para elaborar las tablas lógicas que gobiernan dichos circuitos.

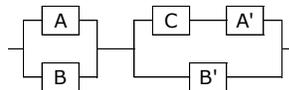
Ejemplo #1: Elabore el polinomio lógico correspondiente al siguiente circuito:



Solución: Se debe RECORDAR que las conexiones en serie se agrupan de dos en dos de izquierda a derecha y las conexiones en paralelo se agrupan de dos en dos de arriba hacia abajo

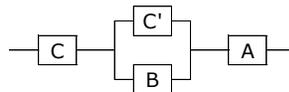
1er paso: Transformar el circuito:

Resultando: $\{(A \vee B) \wedge [(C \wedge A') \vee B']\}$



2do paso: Transformar el circuito:

Resultando: $\{[C \wedge (C' \vee B)] \wedge A\}$



3er paso: Como los circuitos de los pasos 1 y 2 se encuentran conectados en paralelo, simplemente los conectamos con una disyunción colocando primero el circuito del primer paso y después el circuito del segundo paso, resultando finalmente el siguiente polinomio:

$$\{(A \vee B) \wedge [(C \wedge A') \vee B']\} \vee \{[C \wedge (C' \vee B)] \wedge A\}$$

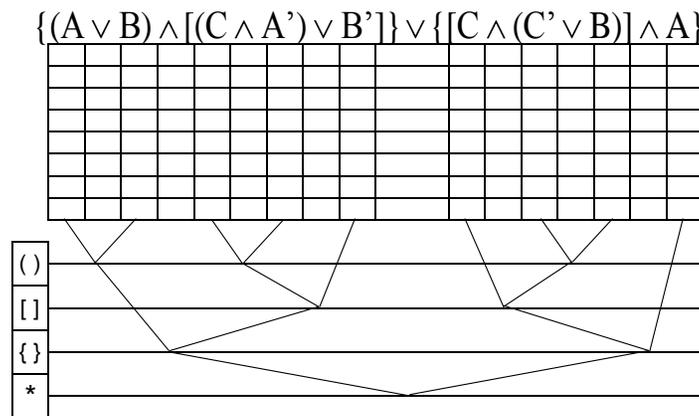
Árbol de programación:

Es un grafo que permite establecer un camino de cómo se construye la tabla lógica de un polinomio, este grafo se dibuja debajo de la tabla lógica y los nodos de dicho árbol se resuelven de acuerdo al siguiente orden de los signos de agrupación: primero los paréntesis (), luego los corchetes [], luego las llaves {}, y así sucesivamente hasta llegar al conectivo principal que es la parte de la tabla que indica la respuesta del ejercicio, dicho conectivo principal es la conjunción (\wedge) o la disyunción (\vee) que agrupa a los factores principales del polinomio. Se suele identificar en el árbol con un asterisco (*).

La cantidad de filas que posee una tabla lógica sigue la fórmula: $\text{filas}=2^n$, donde n es el número de dispositivos involucrados en el polinomio y que se representan con letras del abecedario en mayúsculas, llamadas variables, por ejemplo si en un polinomio se involucran dos variables o dispositivos (A y B), su cantidad de filas sería $2^2=4$; si son tres (A, B y C) su cantidad de filas sería $2^3=8$. En general, se tendrá que $2^n=k$ siendo k un número par que indica cuántas filas han de conformar la tabla lógica del polinomio. La cantidad de columnas en una tabla lógica no se calcula por fórmulas sino que por cada dispositivo o variable se dibuja debajo de ella una columna, al hacer esto automáticamente quedan definidas las columnas de los conectivos.

En el circuito del ejemplo#1, su polinomio lógico es $\{(A \vee B) \wedge [(C \wedge A') \vee B']\} \vee \{[C \wedge (C' \vee B)] \wedge A\}$, construyamos su tabla lógica, incluyendo su árbol de programación.

Solución: la cantidad de filas serían $2^3=8$ y dibujando las columnas debajo de cada letra y conector, la tabla lógica y el árbol de programación sería el siguiente:



Ahora vale la pena saber cómo se debe rellenar (completar) dicha estructura con ceros y unos para realizar el estudio de dicho polinomio. Se sigue el siguiente criterio: Para dar valores a las variables se priorizan en orden ascendente según el alfabeto (de izquierda a derecha: A, B, C,..., Z). Todas y cada una tienen el mismo número de filas en su respectiva columna,

Sabiendo que la disyunción es apagado (0) cuando sus elementos son apagados (0), y la conjunción es encendido (1) cuando sus elementos son encendidos (1), la resolución de los paréntesis en la tabla que nos ocupa es la siguiente: En el nodo $(A \vee B)$, como es una disyunción se escribe cero (0) en la fila donde ambos elementos son cero (0) y eso ocurre en la dos últimas filas; el resto de filas de dicha columna se rellenan con uno (1). En el nodo $(C \wedge A')$ como es una conjunción se escribe uno (1) en la fila donde sus elementos son uno (1), eso ocurre en la 5ª y 7ª filas; las filas restantes de dicha columna se rellenan con cero (0). En el nodo $(C' \vee B)$, por ser una disyunción, se escribe cero (0) en la 3ª y 7ª filas, el resto de las filas de dicha columnas se rellenan con uno (1). La tabla lógica resultante es:

$$\{(A \vee B) \wedge [(C \wedge A') \vee B']\} \vee \{[C \wedge (C' \vee B)] \wedge A\}$$

1	1	1	1	0	0	0	0	1	0	1	1	1
1	1	1	0	0	0	0	0	0	1	1	1	1
1	1	0	1	0	0	1	1	1	0	0	0	1
1	1	0	0	0	0	1	1	0	1	1	0	1
0	1	1	1	1	1	0	0	1	0	1	1	0
0	1	1	0	0	1	0	0	0	1	1	1	0
0	0	0	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	0	1	1	0	0

Luego, resolviendo los nodos correspondientes a los corchetes: $[(C \wedge A') \vee B']$, donde guiándose del árbol de programación es la conjunción del resultado del paréntesis $(C \wedge A')$ y la variable B' , solo se escribe cero (0) en la 1ª, 2ª y 6ª filas; en el resto de filas de dicha columna se escribe uno (1). En el caso del nodo $[C \wedge (C' \vee B)]$, bajo la guía del árbol de programación resulta que es la conjunción de la variable C con el resultado del paréntesis $(C' \vee B)$, solo se escribe uno (1) en 1ª y 5ª filas; el resto de las casillas de esa columna se rellena con cero (0), la tabla lógica resultante de lo que se acaba de explicar es la siguiente:

$$\{(A \vee B) \wedge [(C \wedge A') \vee B']\} \vee \{[C \wedge (C' \vee B)] \wedge A\}$$

1	1	1	1	0	0	0	0	1	1	0	1	1	1
1	1	1	0	0	0	0	0	0	0	1	1	1	1
1	1	0	1	0	0	1	1	1	0	0	0	0	1
1	1	0	0	0	0	1	1	0	0	1	1	0	1
0	1	1	1	1	1	1	0	1	1	0	1	1	0
0	1	1	0	0	1	0	0	0	0	1	1	1	0
0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	1	1	0	0

Continuando con la elaboración de la tabla lógica, los nodos correspondientes a las llaves: $\{(A \vee B) \wedge [(C \wedge A') \vee B']\}$ y $\{[C \wedge (C' \vee B)] \wedge A\}$, se resuelven guiándose en ambos casos en el árbol de programación, para la llave $\{(A \vee B) \wedge [(C \wedge A') \vee B']\}$, se escribe uno (1) en la 3ª, 4ª y 5ª casillas; en

las restantes se rellenan con cero (0). Para la llave $\{[C \wedge (C' \vee B)] \wedge A\}$ solo se escribe uno (1) en la primera casilla y las restantes se rellenan con cero (0), quedando la tabla de la siguiente manera:

$$\{(A \vee B) \wedge [(C \wedge A') \vee B']\} \vee \{[C \wedge (C' \vee B)] \wedge A\}$$

1	1	1	0	1	0	0	0	0	1	1	0	1	1	1	1	
1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	0	1
1	1	0	1	1	0	0	1	1	1	1	0	0	0	0	0	1
1	1	0	1	0	0	0	1	1	1	0	0	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0	0
0	1	1	0	0	0	1	0	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	0

()
[]
{ }
*

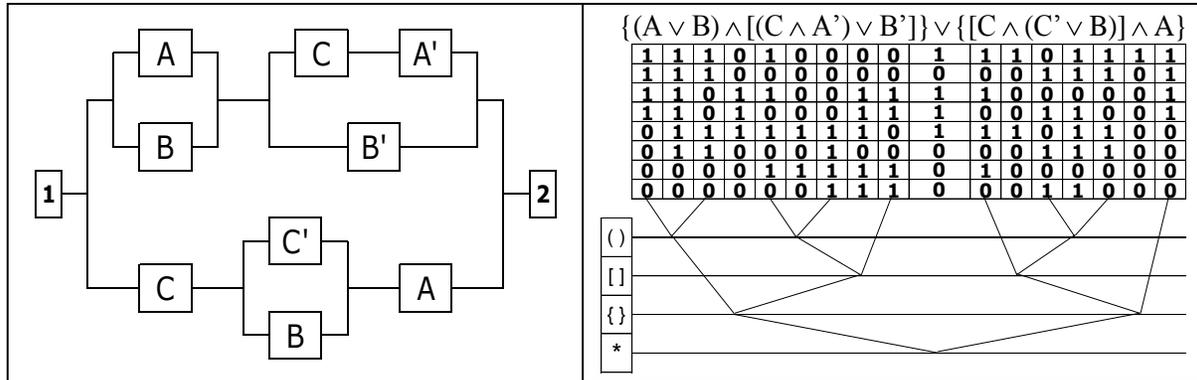
Finalmente, el nodo correspondiente al conectivo principal cuyo nivel de programación está identificado con un asterisco (*) es la conjunción de los resultados de las llaves anteriores, al valorar se escribe cero (0) en la 2ª, 6ª, 7ª y 8ª casillas; mientras que las casillas restantes se llenan con uno (1). Vale la pena acotar que cuando la columna correspondiente al conectivo principal todas sus casillas son cero (0), el polinomio es una CONTRADICCIÓN desde el punto de vista lógico; para efectos del circuito eso es un CORTO CIRCUITO, es decir que ese circuito con esa configuración no va a permitir el flujo de corriente. En cambio, cuando todas las casillas del conectivo principal son uno (1), el polinomio es una TAUTOLOGÍA desde la lógica; pero en el lenguaje del circuito eso es un PUENTE, es decir ese circuito siempre permitirá el paso de corriente desde el punto (1) al punto (2). Pero si las casillas correspondientes a la columna del conectivo principal tienen ceros (0) y unos (1) en distintas combinaciones el polinomio es una CONTINGENCIA y desde el punto de vista del circuito es lo que se denomina un circuito NORMAL. Ya que un circuito normalmente bajo ciertas condiciones permite el flujo de corriente y para otras condiciones no lo permite. Después de echa esta acotación, se presenta la tabla lógica final:

$$\{(A \vee B) \wedge [(C \wedge A') \vee B']\} \vee \{[C \wedge (C' \vee B)] \wedge A\}$$

1	1	1	0	1	0	0	0	0	1	1	0	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	1
1	1	0	1	1	0	0	1	1	1	1	0	0	0	0	1
1	1	0	1	0	0	0	1	1	1	0	0	1	1	0	1
0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0
0	1	1	0	0	0	1	0	0	0	0	1	1	1	0	0
0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0

()
[]
{ }
*

Esta tabla lógica corresponde a una CONTINGENCIA o a un circuito normal, y tiene 4 maneras de funcionar o permitir el paso de corriente y 4 maneras de cerrar dicho paso de corriente. Los colores se usaron con fines didácticos. Se presenta a continuación el circuito y su tabla lógica:



En conclusión: el circuito permite el paso de corriente desde el punto (1) hasta el punto (2) en cualquiera de los siguientes casos:

- 1.- Dispositivos: A (Encendido), B (Encendido), C (Encendido)
- 2.- Dispositivos: A (Encendido), B (Apagado), C (Encendido)
- 3.- Dispositivos: A (Encendido), B (Apagado), C (Apagado)
- 4.- Dispositivos: A (Apagado), B (Encendido), C (Encendido)

Hay diversas maneras de representar los estados del circuito, una de ellas consiste en la elaboración de una tabla de programación auxiliar donde se represente dichos estados, la tabla auxiliar puede ser añadida a la tabla lógica o se puede hacer por separada, en este caso se hará por separado para mostrar su respectivo llenado, ya que se puede llenar como se desee, (bajo los criterios del estado del circuito, o en orden numérico de las variables), pero a efectos prácticos es irrelevante el estilo del llenado y solamente se hará para organizar la información, la cual al ser la misma no altera el contenido de la tabla lógica. Se presenta a continuación un par de ejemplos de tablas de programación auxiliar para el ejemplo que se ha desarrollado hasta ahora:

Tabla organizada por ESTADO del circuito

A	B	C	ESTADO
1	1	1	ENCENDIDO
1	0	1	ENCENDIDO
1	0	0	ENCENDIDO
0	1	1	ENCENDIDO
1	1	0	APAGADO
0	1	0	APAGADO
0	0	1	APAGADO
0	0	0	APAGADO

Tabla organizada por ORDEN de las variables

A	B	C	ESTADO
1	1	1	ENCENDIDO
1	1	0	APAGADO
1	0	1	ENCENDIDO
1	0	0	ENCENDIDO
0	1	1	ENCENDIDO
0	1	0	APAGADO
0	0	1	APAGADO
0	0	0	APAGADO

Conversión del polinomio lógico al circuito serie-paralelo:

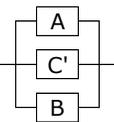
Es posible hacer la conversión de polinomios lógicos a circuitos serie-paralelo, sin embargo la profundización de este tópico se hará en las próximas páginas del presente artículo ya que son extensos y requieren un dominio más que elemental debido al hecho que los polinomios lógicos no poseen exclusivamente conjunciones (\wedge) o disyunciones (\vee) como únicos conectivos lógicos, también poseen condicionales (\Rightarrow) y bicondicionales (\Leftrightarrow). Además que las negaciones en un polinomio no afectan solamente a una variable sino también a estructuras más elaboradas, allí es vital dominar bien las leyes de D'Morgan y similares para lograr una traducción sin errores, por los momentos se abordará la traducción de polinomios lógicos sencillos a circuitos serie-paralelo.

Los polinomios lógicos elementales son aquellos compuestos por conjunciones o disyunciones y las negaciones solamente afectan a una variable, el procedimiento básicamente consiste en transformar desde los signos de agrupación más elementales como los paréntesis () y luego continúa haciendo con los corchetes [] y demás signos de agrupación hasta llegar al conectivo principal.

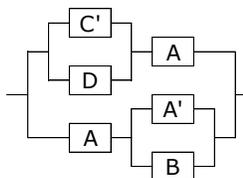
Ejemplo#2: Dibuje el circuito correspondiente al siguiente polinomio:

$$[(A \vee C') \vee B] \wedge \{[(C' \vee D) \wedge A] \vee [A \wedge (A' \vee B)]\}$$

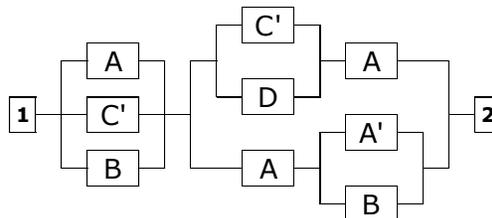
1er paso: La traducción del tramo: $[(A \vee C') \vee B]$ es:



2do paso: La traducción del tramo: $\{[(C' \vee D) \wedge A] \vee [A \wedge (A' \vee B)]\}$ es:



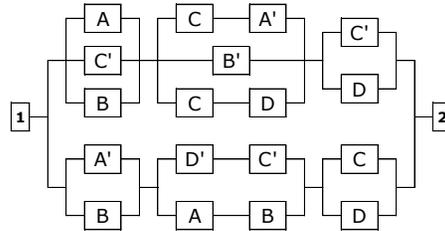
3er paso: El conectivo que une el corchete $[(A \vee C') \vee B]$ con la llave $\{[(C' \vee D) \wedge A] \vee [A \wedge (A' \vee B)]\}$, es una conjunción (\wedge) lo que implica que el circuito dibujado en el primer paso se conecta en serie con el circuito dibujado en el segundo paso, con lo que el circuito resultante es el siguiente:



Hasta aquí se hace el abordaje de la fusión de los contenidos concernientes a álgebra booleana y circuitos serie-paralelo, en las siguientes páginas se profundizará en estos tópicos y

se abordarán otros conectivos lógicos como condicionales y bicondicionales, las leyes de D’Morgan, operaciones del álgebra aplicada a circuitos serie-paralelo, entre otras cosas.

Se dejará al lector como actividad la verificación de la construcción del polinomio lógico correspondiente al siguiente circuito serie-paralelo: (Puede después de ello realizar la tabla lógica)



$$\langle [((A \vee C') \vee B) \wedge \{[(C \wedge A') \vee B] \vee (C \wedge D)\}] \wedge (C' \vee D) \rangle \vee \{ (A' \vee B) \wedge [(D' \wedge C') \vee (A \wedge B)] \} \wedge (C \vee D) |$$

Leyes de D’Morgan en Polinomios Lógicos y su conversión a circuitos serie-paralelo:

Como ya se mencionó, los circuitos serie-paralelo están gobernados en sus traducciones por las conjunciones y disyunciones, pero es hora de profundizar en los aspectos concernientes a la traducción de polinomios lógicos donde las negaciones no solo afecten a las variables sino a las agrupaciones de ellas, ya que se pueden traducir o dibujar conexiones en serie o en paralelo; pero traducir o dibujar la negación de una asociación no es posible ya que estamos hablando del estado contrario de dicha asociación. En esos casos se usará las leyes de D’Morgan, las cuales enuncian lo siguiente:

Primera Ley: $(A \wedge B)' \equiv A' \vee B'$. El complemento o negación de una asociación binaria en serie es equivalente a la asociación en paralelo de las negaciones o complementos de sus elementos. A continuación sus tablas lógicas donde se constata la veracidad de dicha ley:

$(A \wedge B)'$			$A' \vee B'$		
1	0	1	0	0	0
1	1	0	0	1	1
0	1	1	1	1	0
0	1	0	1	1	1

Segunda Ley: $(A \vee B)' \equiv A' \wedge B'$. El complemento o negación de una asociación binaria en paralelo es equivalente a la asociación en serie de las negaciones o complementos de sus elementos. A continuación sus tablas lógicas donde se constata la veracidad de dicha ley:

$(A \vee B)'$			$A' \wedge B'$		
1	0	1	0	0	0
1	0	0	0	0	1
0	0	1	1	0	0
0	1	0	1	1	1

Además de este par de leyes muy útiles se debe añadir una propiedad que se deriva de la doble negación, a la cual se le llama idempotencia de la negación, cuya fórmula es: $(A')' \equiv A$.

Con estos elementos ahora es posible traducir polinomios lógicos donde las negaciones o complementos que afecten a signos de agrupación $()$, $[]$, $\{\}$, $||$, $\langle \rangle$, $||$, etc. Para verificar si la traducción es correcta se harán las tablas lógicas del polinomio original y del polinomio transformado y se debe verificar que los conectivos principales de ambos polinomios deben ser exactamente iguales.

Ejemplo: Dado el siguiente polinomio:

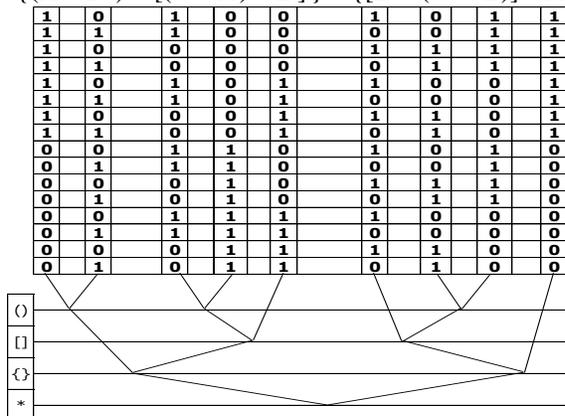
$$\{(A \wedge D')' \wedge [(C \wedge A') \vee B']\}' \wedge \{[D \wedge (C' \vee B)] \wedge A\}'$$

Elabore su tabla lógica, el polinomio lógico equivalente con su respectiva tabla y el circuito serie-paralelo correspondiente.

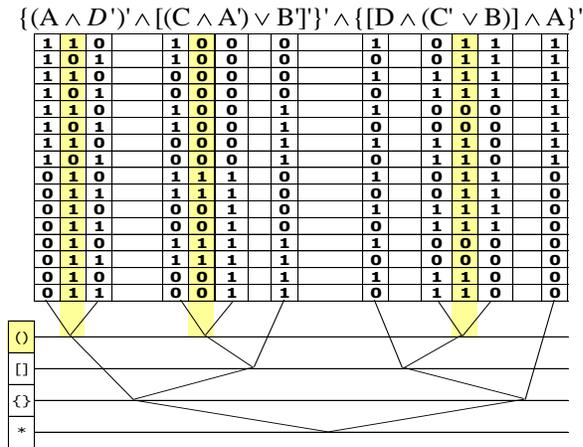
Solución: El trabajo se organiza en 4 etapas, la primera etapa, consiste en construir la tabla lógica de dicho polinomio. La segunda etapa, consiste en aplicar las leyes de los complementos o negaciones (D'Morgan e idempotencia) donde corresponda hasta lograr que las negaciones afecten solamente a las variables del polinomio, La tercera etapa consiste en elaborar la tabla lógica del polinomio que resultó en la segunda etapa y si sus conectivos principales son iguales se procede con la cuarta etapa en la cual se elabora el dibujo de dicho circuito.

Primera Etapa: Tabla lógica. De manera didáctica esta tabla se hará con la ayuda de su árbol de programación, como este polinomio tiene 4 variables, entonces su cantidad de filas es $2^4=16$, la primera variable (A), tendría la secuencia definida por 8 casillas encendidas y 8 casillas apagadas, (A') tendría la secuencia definida por 8 casillas apagadas seguida de 8 casillas encendidas, la variable (B) tendría la secuencia 4 encendidas seguida de 4 apagados, (B') tendría la secuencia 4 apagados seguidos de 4 encendidos, la variable (C) tendría la secuencia 2 encendidas seguida de 2 apagados, la variable (C') tendría la secuencia 2 apagadas seguida de 2 encendidos y finalmente la variable (D) tendría la secuencia de 1 encendido seguido de 1 apagado, la variable (D') tendría la secuencia 1 apagado seguido de 1 encendido:

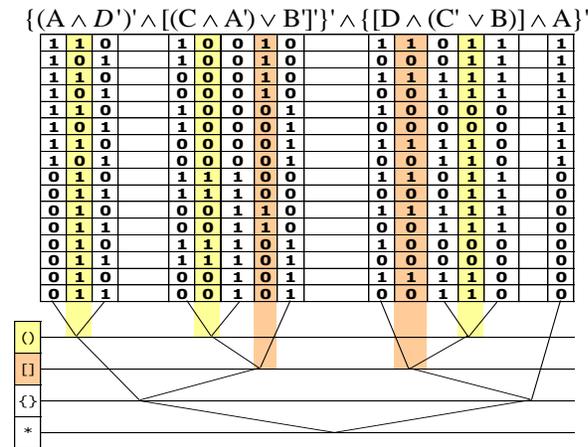
$$\{(A \wedge D')' \wedge [(C \wedge A') \vee B']\}' \wedge \{[D \wedge (C' \vee B)] \wedge A\}'$$



Se procede a resolver los paréntesis, el primer paréntesis $(A \wedge D)'$ es el complemento o negación de una conjunción, esto quiere decir que donde dicho paréntesis se cumple la ley de la conjunción que se debería escribir (1) pero por culpa de la negación de dicho paréntesis no se puede escribir (1) sino (0), eso ocurre en las casillas 2°, 4°, 6° y 8°; las restantes casillas de esa columna se escribe (1). El paréntesis $(C \wedge A)$ se resuelve como una conjunción normal ya que no se encuentra negada es decir se escribe (1) en las casillas 9°, 10°, 13° y 14°; las restantes casillas de esa columna se escribe (0). El paréntesis $(C' \vee B)$ es una disyunción normal, es decir se escribe (0) en las casillas 5°, 6°, 13°, 14°; las restantes casillas de esa columna se escribe (1):

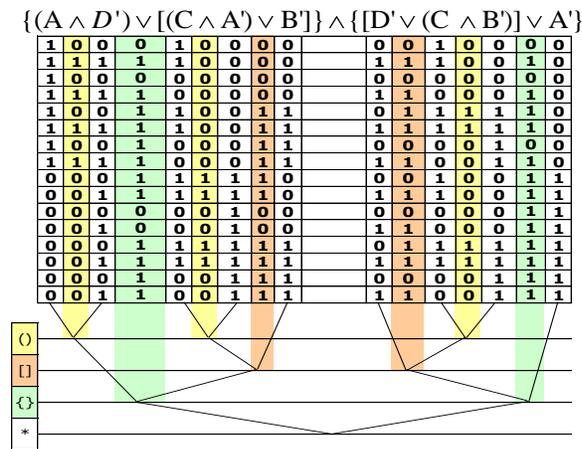


Continuando con la resolución de los corchetes, el primer corchete $[(C \wedge A) \vee B']'$ es el complemento o negación de la conjunción del resultado del paréntesis $(C \wedge A)$ y la variable B' , es decir donde se cumpla la ley de la disyunción para este corchete en vez de escribir (0) se escribe (1) y eso ocurre en las casillas 1°, 2°, 3°, 4°, 11° y 12°; en el resto de las casillas de dicha columna se escribe (0). En el segundo corchete $[D \wedge (C' \vee B)]$ es la conjunción de la variable D y el resultado del paréntesis $(C' \vee B)$ es decir se escribe (1) en las casillas 2°, 4°, 8°, 10°, 12°, 16°; en el resto de las casillas de dicha columna se escribe (0):

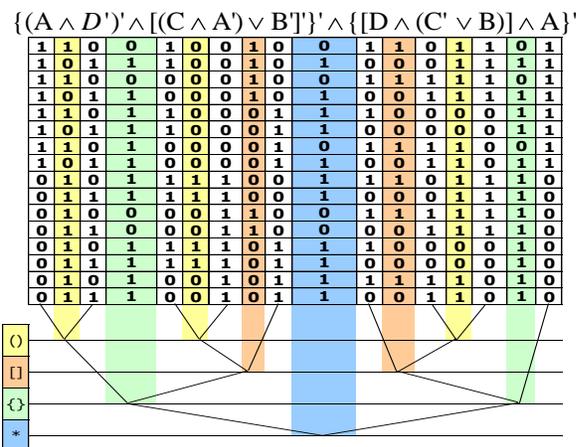


Procediendo a resolver las llaves: $\{(A \wedge D)'\} \wedge \{(C \wedge A) \vee B'\}'$ es la negación o complemento de la conjunción que une el resultado del paréntesis $(A \wedge D)'$ y el corchete $[(C \wedge A) \vee B']'$, es decir en vez

de escribir (1) en donde se cumple la ley de la conjunción se escribe (0), eso ocurre en las casillas 1º, 3º, 11º, 12º; en el resto de las casillas de dicha columna se escribe (1). La segunda llave $\{[D \wedge (C \vee B)] \wedge A\}'$ es la negación o complemento de la conjunción que une el resultado del corchete $[D \wedge (C \vee B)]$ y la variable A, es decir en vez de escribir (1) en donde se cumple la ley de la conjunción se escribe (0), eso ocurre en las casillas 2º, 4º y 8º; en el resto de las casillas de esa columna se escribe (1):



Al resolver el conectivo principal, el cual es la conjunción entre los resultados de las llaves $\{(A \wedge D') \vee [(C \wedge A) \vee B']\}'$ y $\{[D \vee (C \wedge B)] \vee A\}'$, los conectivos principales de los polinomios NUNCA son afectados por negaciones o complementos, en este caso simplemente se buscará donde se cumpla la ley de la conjunción señalada es decir se escribe (1) en las casillas 2º, 4º, 5º, 6º, 8º, 9º, 10º, 13º, 14º, 15º, 16º; en el resto de las casillas de dicha columna se escribe (0):



En la siguiente tabla se añade los estados del circuito a la parte derecha y se basan en los resultados de la columna sombreada en azul que corresponde al conectivo principal, no se incluyó al árbol de programación ya que dicho árbol es una ayuda didáctica para novales en estos menesteres, en la parte inferior se incluye la codificación de la tabla, la cual es la que vamos a comparar con la codificación de la tabla transformada, si coinciden exactamente entonces el ejercicio está realizado de manera correcta:

$$\{(A \wedge D)' \wedge [(C \wedge A) \vee B']\}' \wedge \{[D \wedge (C' \vee B)] \wedge A\}'$$

1	1	0	0	1	0	0	1	0	0	1	1	0	1	1	0	1	APAGADO
1	0	1	1	1	0	0	1	0	1	0	0	0	1	1	1	1	ENCENDIDO
1	1	0	0	0	0	0	1	0	0	1	1	1	1	1	1	0	APAGADO
1	0	1	1	0	0	0	1	0	1	0	0	1	1	1	1	1	ENCENDIDO
1	1	0	1	1	0	0	0	1	1	1	0	0	0	0	0	1	ENCENDIDO
1	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	1	ENCENDIDO
1	1	0	1	0	0	0	0	1	0	1	1	1	1	0	0	1	APAGADO
1	0	1	1	0	0	0	0	1	1	0	0	1	1	0	1	1	ENCENDIDO
0	1	0	1	1	1	1	0	0	1	1	1	0	1	1	1	0	ENCENDIDO
0	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	0	ENCENDIDO
0	1	0	0	0	0	1	1	0	0	1	1	1	1	1	1	0	APAGADO
0	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	APAGADO
0	1	0	1	1	1	1	0	1	1	0	0	0	0	0	1	0	ENCENDIDO
0	1	1	1	1	1	1	0	1	1	0	0	0	0	0	1	0	ENCENDIDO
0	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	1	ENCENDIDO
0	1	1	1	0	0	1	0	1	1	1	1	1	0	1	0	1	ENCENDIDO

Codificación de la tabla: (Se toma del conectivo principal de arriba hacia abajo y se escribe de izquierda a derecha)

0	1	0	1	1	1	0	1	1	1	0	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Segunda Etapa: Polinomio transformado. (Se aplican las leyes del complemento desde los signos de agrupación más grandes hacia los más pequeños, ya que de esta manera es más eficiente la detección de idempotencias que permiten cancelarlas)

- $\{(A \wedge D)' \wedge [(C \wedge A) \vee B']\}' \wedge \{[D \wedge (C' \vee B)] \wedge A\}'$ Polinomio Original
- $\{(A \wedge D')' \vee [(C \wedge A') \vee B']\}'' \wedge \{[D \wedge (C' \vee B)]' \vee A'\}$ Aplicación de D'Morgan en las llaves
- $\{(A \wedge D') \vee [(C \wedge A') \vee B']\} \wedge \{[D \wedge (C' \vee B)]' \vee A'\}$ Aplicación de la idempotencia dentro de la 1º llave
- $\{(A \wedge D') \vee [(C \wedge A') \vee B']\} \wedge \{[D' \vee (C' \vee B)'] \vee A'\}$ Aplicación de D'Morgan en el 2º corchete
- $\{(A \wedge D') \vee [(C \wedge A') \vee B']\} \wedge \{[D' \vee (C'' \wedge B')] \vee A'\}$ Aplicación de D'Morgan en el 3º paréntesis
- $\{(A \wedge D') \vee [(C \wedge A') \vee B']\} \wedge \{[D' \vee (C \wedge B')] \vee A'\}$ Aplicación de la idempotencia dentro del 3º paréntesis

Se concluye que el polinomio transformado es: $\{(A \wedge D') \vee [(C \wedge A') \vee B']\} \wedge \{[D' \vee (C \wedge B')] \vee A'\}$

Tercera Etapa: Tabla lógica del polinomio transformado. Al igual que la tabla lógica del polinomio original, con fines didácticos, esta tabla se hará con la ayuda de su árbol de programación, como este polinomio tiene 4 variables, entonces su cantidad de filas es $2^4=16$, la primera variable (A), tendría la secuencia definida por 8 casillas encendidas y 8 casillas apagadas, (A') tendría la secuencia definida por 8 casillas apagadas seguida de 8 casillas encendidas, la variable (B) tendría la secuencia 4 encendidas seguida de 4 apagados, (B') tendría la secuencia 4 apagados seguidos de 4 encendidos, la variable (C) tendría la secuencia 2 encendidas seguida de 2 apagados, la variable (C') tendría la secuencia 2 apagadas seguida de 2 encendidos y finalmente la variable (D) tendría la secuencia de 1 encendido seguido de 1 apagado, la variable (D') tendría la secuencia 1 apagado seguido de 1 encendido:

$$\{(A \wedge D') \vee [(C \wedge A') \vee B']\} \wedge \{[D' \vee (C \wedge B')] \vee A'\}$$

1	0	1	0	0	0	1	0	0
1	1	1	1	0	0	1	1	0
1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0
1	0	1	0	1	0	1	1	0
1	1	1	0	1	1	1	1	0
1	0	0	0	1	0	0	1	0
1	1	0	0	1	1	0	1	0
0	0	1	1	0	0	1	1	1
0	1	1	1	0	1	1	0	1
0	0	0	1	0	0	0	0	1
0	1	0	1	0	1	0	0	1
0	0	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1
0	0	0	1	1	0	0	1	1
0	1	0	1	1	1	0	1	1

0

1

0

1

0

1

0

1

0

1

Resolviendo los paréntesis: En $(A \wedge D')$, se cumple la conjunción en las casillas 2°, 4°, 6° y 8°, se escribe (1) en ellas; en el resto se escribe (0). En $(C \wedge A')$, se cumple la conjunción en las casillas 9°, 10°, 13° y 14°, se escribe (1) en ellas; en el resto se escribe (0). En $(C \wedge B')$, se cumple la conjunción en las casillas 5°, 6°, 13° y 14°, se escribe (1) en ellas; en el resto se escribe (0):

$$\{(A \wedge D') \vee [(C \wedge A') \vee B']\} \wedge \{[D' \vee (C \wedge B')] \vee A'\}$$

1	0	0	1	0	0	0	0	1	0	0	0
1	1	1	1	0	0	0	0	1	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	1	0	0	0
1	0	0	1	0	0	1	0	1	1	1	0
1	1	1	1	0	0	1	1	1	1	1	0
1	0	0	0	0	0	1	0	0	0	1	0
1	1	1	0	0	0	1	1	0	0	1	0
0	0	0	1	1	1	0	0	1	0	0	1
0	0	1	1	1	1	0	1	1	0	0	1
0	0	0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	1	0	1	0	0	0	1
0	0	0	1	1	1	1	0	0	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	0	0	1	1
0	0	1	0	0	1	1	1	0	0	1	1

0

1

0

1

0

1

0

1

0

1

Resolviendo los corchetes: En $[(C \wedge A') \vee B']$, se cumple las leyes de la disyunción en las casillas 1° al 4°, 11°, 12° es decir allí se escribe (0); en el resto se escribe (1), En el segundo corchete $[D' \vee (C \wedge B')]$, se cumple las leyes de la disyunción en las casillas 1°, 3°, 7°, 9°, 11°, 15°, allí se escribe (0) y en el resto de las casillas de dicha columna se escribe (1):

$$\{(A \wedge D') \vee [(C \wedge A') \vee B']\} \wedge \{[D' \vee (C \wedge B')] \vee A'\}$$

1	0	0	1	0	0	0	0	0	1	0	0
1	1	1	1	0	0	0	0	1	1	1	0
1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	1	1	0	0
1	0	0	1	0	0	1	1	0	1	1	1
1	1	1	1	0	0	1	1	1	1	1	0
1	0	0	0	0	0	1	1	0	0	0	1
1	1	1	0	0	0	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	1	0	0
0	0	1	1	1	1	0	0	1	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	1	0	0	1	1	0	0
0	0	0	1	1	1	1	1	0	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	0	0	0	1
0	0	1	0	0	1	1	1	1	0	0	1

0

1

0

1

0

1

0

1

0

1

$$\{(A \wedge D') \vee [(C \wedge A) \vee B']\} \wedge \{[D' \vee (C \wedge B')] \vee A\}$$

1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	APAGADO
1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	1	ENCENDIDO
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	APAGADO
1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	1	ENCENDIDO
1	0	0	1	1	0	0	1	1	1	0	1	1	1	1	1	ENCENDIDO
1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	ENCENDIDO
1	0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	APAGADO
1	1	1	1	0	0	0	1	1	1	1	0	0	1	1	0	ENCENDIDO
0	0	0	1	1	1	1	1	0	1	0	0	1	0	0	1	ENCENDIDO
0	0	1	1	1	1	1	1	0	1	1	1	1	0	0	1	ENCENDIDO
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	APAGADO
0	0	1	0	0	0	1	0	0	0	1	1	0	0	0	1	APAGADO
0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	ENCENDIDO
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	ENCENDIDO
0	0	0	1	0	0	1	1	1	1	1	0	0	0	0	1	ENCENDIDO
0	0	1	1	0	0	1	1	1	1	1	0	0	1	1	1	ENCENDIDO

Codificación de la tabla transformada: (Se toma del conector principal de arriba hacia abajo y se escribe de izquierda a derecha)

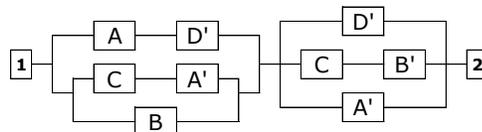
0	1	0	1	1	1	0	1	1	1	0	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Se procede a comparar las codificaciones de ambas tablas y deben coincidir los valores en las mismas posiciones:

Tabla Original:	0	1	0	1	1	1	0	1	1	1	0	0	1	1	1	1
Tabla Transformada:	0	1	0	1	1	1	0	1	1	1	0	0	1	1	1	1
Verificación:	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Como se puede apreciar, Ambas tablas lógicas son EQUIVALENTES.

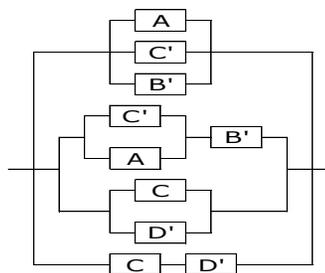
Cuarta Etapa: Circuito serie-paralelo correspondiente. (Se hace a partir de la traducción del polinomio transformado)



Hasta aquí el abordaje de las leyes de D’Morgan o leyes de las negaciones o complementos, en las siguientes páginas se hará el abordaje y profundización de la transformación de polinomios lógicos con conectivos condicionales y bicondicionales a circuitos serie-paralelo. Se deja al lector la verificación del siguiente ejercicio:

- Polinomio: $\langle [[(A \vee C') \vee B']' \wedge \{[(C \wedge A') \vee B] \vee (C' \wedge D) \}' \wedge (C' \vee D)]' \rangle$

- Circuito serie-paralelo correspondiente:



Polinomios lógicos con condicionales y bicondicionales y su conversión a circuitos serie-paralelo:

Como ya se mencionó, los circuitos serie-paralelo están gobernados en sus traducciones por las conjunciones y disyunciones, donde las conjunciones (\wedge) representan series y las disyunciones (\vee) representan paralelos, en páginas anteriores se abordaron los fundamentos de las negaciones a través del manejo de las leyes de D'Morgan y las idempotencias, es el momento de profundizar en los aspectos concernientes a la traducción de polinomios lógicos que contienen conectivos de condicionales (\rightarrow) y bicondicionales (\leftrightarrow). Como ya se sabe, traducir o dibujar la negación de una asociación no es posible ya que estamos hablando del estado contrario de dicha asociación.

En el caso de los condicionales y bicondicionales, tampoco se puede traducir o dibujar porque no se sabe que representa un condicional y un bicondicional en términos de series y paralelos. Por ello es fundamental, establecer las leyes que corresponden para los condicionales y bicondicionales.

Ley del condicional: $A \rightarrow B \equiv A' \vee B$. La conexión condicional entre dos dispositivos es equivalente a la disyunción de la negación del primer dispositivo con el segundo. A continuación sus tablas lógicas donde se constata la veracidad de dicha ley: (El condicional no permite el flujo de corriente si el primer dispositivo está encendido y el segundo está apagado)

$A \rightarrow B$		
1	1	1
1	0	0
0	1	1
0	1	0

$A' \vee B$		
0	1	1
0	0	0
1	1	1
1	1	0

Ley del bicondicional: $A \leftrightarrow B \equiv (A' \vee B) \wedge (B' \vee A)$. La conexión bicondicional entre los dispositivos A y B es la conjunción de la disyunción entre la negación del primer dispositivo con el segundo y la disyunción entre la negación del segundo dispositivo con el primer dispositivo. A continuación sus tablas lógicas donde se constata la veracidad de dicha ley: (El bicondicional permite el flujo de corriente si el estado de ambos dispositivos son iguales)

$A \leftrightarrow B$		
1	1	1
1	0	0
0	0	1
0	1	0

$(A' \vee B) \wedge (B' \vee A)$						
0	1	1	1	0	1	1
0	0	0	0	1	1	1
1	1	1	0	0	0	0
1	1	0	1	1	1	0

Además de las leyes del condicional y bicondicional, se enunciarán las propiedades correspondientes a las negaciones del condicional y bicondicional:

$$(A \rightarrow B)' \equiv A \wedge B'$$

$$(A \leftrightarrow B)' \equiv (A \wedge B') \vee (B \wedge A')$$

Con estos elementos ahora es posible traducir polinomios lógicos que posean conexiones condicionales o bicondicionales. Para verificar si la traducción es correcta, se procederá de forma análoga a lo que se describió en las páginas anteriores, se harán las tablas lógicas del polinomio original y del polinomio transformado y se debe verificar que los conectivos principales de ambos polinomios deben ser exactamente iguales.

Vale la pena destacar que el equivalente del bicondicional tiene la propiedad de expandir el polinomio lógico resultante, si consideramos que las variables solas son polinomios atómicos y las agrupaciones de variables en paréntesis, corchetes, llaves, etc, son polinomios moleculares entonces si un paréntesis contiene un bicondicional se convierte en un corchete, si un corchete contiene un bicondicional se convierte en una llave, y así sucesivamente, por lo tanto es natural que los polinomios equivalentes a polinomios que contienen bicondicionales “crezcan” o se “expandan”, por lo tanto no hay motivo para pensar que se está obrando de manera incorrecta, ya que al fin y al cabo, el criterio que guía las conversiones de dichos polinomios son los conectivos principales de sus tablas lógicas. Dicho esto, se procede a mostrar un ejemplo para aplicar los fundamentos recientemente explicados.

Ejemplo: Dado el siguiente polinomio:

$$\{(A \leftrightarrow B) \wedge [(C \leftrightarrow A) \vee (B \vee C)]\}' \wedge [(D \rightarrow C)' \leftrightarrow (D' \wedge B)]$$

Elabore su tabla lógica, el polinomio lógico equivalente con su respectiva tabla y el circuito serie-paralelo correspondiente.

Solución: El trabajo se organiza en 4 etapas, la primera etapa, consiste en construir la tabla lógica de dicho polinomio. La segunda etapa, consiste en aplicar las leyes correspondientes (D’Morgan, idempotencia, condicionales y bicondicionales) hasta lograr que las negaciones afecten solamente a las variables del polinomio y sus conectivos sean conjunciones y disyunciones, La tercera etapa consiste en elaborar la tabla lógica del polinomio que resultó en la segunda etapa y si sus conectivos principales o codificaciones principales son iguales se procede con la cuarta etapa en la cual se elabora el dibujo de dicho circuito.

Primera Etapa: Tabla lógica. De manera didáctica esta tabla se hará con la ayuda de su árbol de programación, como este polinomio tiene 4 variables, entonces su cantidad de filas es $2^4=16$, la primera variable (A), tendría la secuencia definida por 8 casillas encendidas y 8 casillas apagadas, (A’) tendría la secuencia definida por 8 casillas apagadas seguida de 8 casillas encendidas, la variable (B) tendría la secuencia 4 encendidas seguida de 4 apagados, (B’) tendría la secuencia 4 apagados seguidos de 4 encendidos, la variable (C) tendría la secuencia 2 encendidas seguida de 2 apagados, la variable (C’) tendría la secuencia 2 apagadas seguida de 2 encendidos y finalmente la variable (D) tendría la secuencia de 1 encendido seguido de 1 apagado, la variable (D’) tendría la secuencia 1 apagado seguido de 1 encendido:

$$\{(A \leftrightarrow B)'\} \wedge [(C' \leftrightarrow A) \vee (B \vee C)'] \wedge [(D \rightarrow C)' \leftrightarrow (D' \wedge B)]$$

1	0	0	0	1	0	1	1	0	1
1	0	0	0	1	0	0	1	1	1
1	0	1	0	1	1	1	0	0	1
1	0	1	0	1	1	0	0	1	1
1	1	0	0	0	0	1	1	0	0
1	1	0	0	0	0	1	0	1	0
1	1	1	0	0	1	0	1	0	0
1	1	1	0	0	1	0	1	0	0
0	0	0	1	1	0	1	1	0	1
0	0	0	1	1	0	0	1	1	1
0	0	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	0	0	1
0	1	0	1	0	0	1	1	0	0
0	1	0	1	0	0	0	1	1	0
0	1	1	1	0	1	1	0	0	0
0	1	1	1	0	1	0	0	1	0

0

1

0

1

*

Resolviendo los paréntesis: En $(A \leftrightarrow B)'$, es el complemento del bicondicional, esto quiere decir que donde se cumple la ley del bicondicional no se escribe (1) sino (0) en las casillas 5° a la 12°; en el resto se escribe (1). En $(C' \leftrightarrow A)$, es un bicondicional, se escribe (1) en las casillas 1°, 2°, 5°, 6°, 11°, 12°, 15° y 16°; en el resto se escribe (0). En $(D \rightarrow C)'$, es la negación del condicional, es decir donde se cumpla la ley del condicional no se escribe (0) sino (1) en las casillas 1°, 2°, 4°, 5°, 6°, 8°, 9°, 10°, 12°, 13°, 14° y 16°; en el resto se escribe (0). En $(D' \wedge B)$, es una conjunción, se escribe (1) en las casillas 2°, 4°, 10° y 12°; en el resto se escribe (0):

$$\{(A \leftrightarrow B)'\} \wedge [(C' \leftrightarrow A) \vee (B \vee C)'] \wedge [(D \rightarrow C)' \leftrightarrow (D' \wedge B)]$$

1	1	0	0	1	0	1	1	0	1	0	0	1
1	1	0	0	1	0	1	1	0	0	1	1	1
1	1	0	1	0	0	1	1	1	1	0	0	1
1	1	0	1	0	0	1	1	1	0	0	0	1
1	0	1	0	1	0	0	0	0	1	0	1	0
1	0	1	0	1	0	0	0	0	0	0	1	0
1	0	1	1	0	0	0	1	1	1	0	0	0
1	0	1	1	0	0	0	1	1	0	0	0	1
0	0	0	0	0	1	1	1	0	1	0	1	0
0	0	0	0	0	1	1	1	0	0	0	1	1
0	0	0	1	1	1	1	1	1	1	0	0	1
0	0	0	1	1	1	1	1	1	0	0	0	1
0	1	1	0	0	1	0	0	0	1	0	1	0
0	1	1	0	0	1	0	0	0	0	0	1	0
0	1	1	1	1	1	0	1	1	1	0	0	0
0	1	1	1	1	1	0	1	1	0	0	1	0

0

1

0

1

*

Resolviendo los corchetes: En $[(C' \leftrightarrow A) \vee (B \vee C)']$, es una conjunción, se escribe (1) en todas las casillas excepto la 13° y 14° donde se escribe (0). En $[(D \rightarrow C)' \leftrightarrow (D' \wedge B)]$, es el bicondicional, se escribe (1) en las casillas 1°, 5°, 6°, 8°, 9°, 13°, 14° y 16°; en el resto se escribe (0):

$$\{(A \leftrightarrow B)'\} \wedge [(C' \leftrightarrow A) \vee (B \vee C)'] \wedge [(D \rightarrow C)' \leftrightarrow (D' \wedge B)]$$

1	1	0	0	1	0	1	1	1	0	1	0	0	1
1	1	0	0	1	0	1	1	1	0	0	1	0	1
1	1	0	1	0	0	1	1	1	1	1	0	0	0
1	1	0	1	0	0	1	1	1	1	0	0	0	1
1	0	1	0	1	0	1	0	0	0	1	0	0	0
1	0	1	0	1	0	1	0	0	0	0	1	1	0
1	0	1	1	0	0	1	0	1	1	1	0	0	0
1	0	1	1	0	0	1	0	1	1	0	0	0	1
0	0	0	0	0	1	1	1	1	0	1	0	1	0
0	0	0	0	0	1	1	1	1	0	0	0	1	1
0	0	0	1	1	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	0	0	0	1	1
0	1	1	0	0	1	0	0	0	0	1	0	0	0
0	1	1	0	0	1	0	0	0	0	0	1	1	0
0	1	1	1	1	1	0	1	1	1	1	0	0	0
0	1	1	1	1	1	0	1	1	0	0	1	1	0

0

1

0

1

*

Resolviendo la llave $\{(A \leftrightarrow B')' \wedge [(C' \leftrightarrow A) \vee (B \vee C')]\}'$, es la negación de la conjunción, es decir en vez de escribir (1) se escribe (0) en las casillas 1° a la 4°, 15° y 16°; en el resto se escribe (1):

$$\{(A \leftrightarrow B')' \wedge [(C' \leftrightarrow A) \vee (B \vee C')]\}' \wedge [(D \rightarrow C)' \leftrightarrow (D' \wedge B)]$$

1	1	0	0	0	1	0	1	1	1	0	0	1	0	1	1	0	0	1
1	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	1	1
1	1	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0	0	1
1	1	0	0	1	0	0	1	1	1	1	1	0	0	0	0	1	1	1
1	0	1	1	0	1	0	1	0	0	0	0	1	0	1	1	0	0	0
1	0	1	1	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0
1	0	1	1	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0
1	0	1	1	1	0	0	1	0	1	1	1	0	0	0	1	1	0	0
0	0	0	1	0	0	1	1	1	1	0	0	1	0	1	1	0	0	1
0	0	0	1	0	0	1	1	1	1	0	0	0	1	0	1	0	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1
0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1
0	1	1	1	0	0	1	0	0	0	0	0	1	0	1	1	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0
0	1	1	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0
0	1	1	0	1	1	1	1	0	1	1	0	0	0	1	1	0	0	0

0
1
0
*

Al resolver el conectivo principal, el cual es la conjunción se escribe (1) en las casillas 5°, 6°, 8°, 9°, 13°, 14°; en el resto de las casillas de dicha columna se escribe (0):

$$\{(A \leftrightarrow B')' \wedge [(C' \leftrightarrow A) \vee (B \vee C')]\}' \wedge [(D \rightarrow C)' \leftrightarrow (D' \wedge B)]$$

1	1	0	0	0	1	0	1	1	1	0	0	1	0	1	1	0	0	1
1	1	0	0	0	1	0	1	1	1	0	0	0	0	1	0	1	1	1
1	1	0	0	1	0	0	1	1	1	1	0	1	1	0	0	0	0	1
1	1	0	0	1	0	0	1	1	1	1	0	0	0	0	0	1	1	1
1	0	1	1	0	1	0	1	0	0	0	1	1	0	1	1	0	0	0
1	0	1	1	0	1	0	1	0	0	0	1	0	0	1	1	1	0	0
1	0	1	1	1	0	0	1	0	1	1	0	1	1	0	0	0	0	0
1	0	1	1	1	0	0	1	0	1	1	1	0	0	0	1	1	0	0
0	0	0	1	0	0	1	1	1	1	0	1	1	0	1	1	0	0	1
0	0	0	1	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1
0	0	0	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	1
0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
0	1	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	0
0	1	1	1	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0
0	1	1	0	1	1	1	1	0	1	1	0	1	1	0	0	0	0	0
0	1	1	0	1	1	1	1	0	1	1	0	0	0	1	1	0	0	0

0
1
0
*

En la siguiente tabla se añade los estados del circuito a la parte derecha y se basan en los resultados de la columna sombreada en azul que corresponde al conectivo principal y en la parte inferior se incluye la codificación de la tabla, la cual es la que vamos a comparar con la codificación de la tabla transformada, si coinciden exactamente entonces el ejercicio está realizado de manera correcta:

$$\{(A \leftrightarrow B')' \wedge [(C' \leftrightarrow A) \vee (B \vee C')]\}' \wedge [(D \rightarrow C)' \leftrightarrow (D' \wedge B)]$$

1	1	0	0	0	1	0	1	1	1	0	0	1	0	1	1	0	0	1	APAGADO
1	1	0	0	0	1	0	1	1	1	0	0	0	0	1	0	1	1	1	APAGADO
1	1	0	0	1	0	0	1	1	1	1	0	0	0	0	0	1	1	1	APAGADO
1	0	1	1	0	1	0	1	0	0	0	1	1	0	1	1	0	0	0	ENCENDIDO
1	0	1	1	0	1	0	1	0	0	0	1	0	0	1	1	0	0	0	ENCENDIDO
1	0	1	1	1	0	0	1	0	1	1	0	1	1	0	0	0	0	0	APAGADO
1	0	1	1	1	0	0	1	0	1	1	1	0	0	0	1	1	0	0	ENCENDIDO
0	0	0	1	0	0	1	1	1	1	0	1	1	0	1	1	0	0	1	ENCENDIDO
0	0	0	1	0	0	1	1	1	1	0	0	0	0	0	1	1	1	1	APAGADO
0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	APAGADO
0	1	1	1	0	0	1	0	0	0	0	1	1	0	1	1	0	0	0	ENCENDIDO
0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	ENCENDIDO
0	1	1	0	1	1	1	1	0	1	1	0	1	1	0	0	0	0	0	APAGADO
0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	1	0	0	APAGADO

Codificación de la tabla:

0	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Segunda Etapa: Polinomio transformado. (Primero se aplican las leyes del condicional y bicondicional desde los signos de agrupación menores a los mayores, luego se aplican las leyes del complemento desde los signos de agrupación más grandes hacia los más pequeños)

Polinomio Original:

$$\{(A \leftrightarrow B)' \wedge [(C' \leftrightarrow A') \vee (B \vee C')]\}' \wedge [(D \rightarrow C)' \leftrightarrow (D' \wedge B)]$$

Aplicando las leyes del bicondicional y condicional en los paréntesis:

$$|[(A' \vee B') \wedge (B \vee A)]' \wedge \{[(C \vee A') \wedge (A \vee C')] \vee (B \vee C')\}' | \wedge [(D' \vee C)' \leftrightarrow (D' \wedge B)]$$

Aplicando las leyes del bicondicional en el tercer corchete:

$$|[(A' \vee B') \wedge (B \vee A)]' \wedge \{[(C \vee A') \wedge (A \vee C')] \vee (B \vee C')\}' | \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D' \wedge B)' \vee (D' \vee C)']\}$$

Aplicando las leyes de D'Morgan en la barra:

$$|[(A' \vee B') \wedge (B \vee A)] \vee \{[(C \vee A') \wedge (A \vee C')] \vee (B \vee C')\}' | \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D' \wedge B)' \vee (D' \vee C)']\}$$

Aplicando las leyes de D'Morgan en la primera llave:

$$|[(A' \vee B') \wedge (B \vee A)] \vee \{[(C \vee A') \wedge (A \vee C')] \wedge (B \vee C')\}' | \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D' \wedge B)' \vee (D' \vee C)']\}$$

Aplicando las leyes de D'Morgan en el segundo corchete:

$$|[(A' \vee B') \wedge (B \vee A)] \vee \{[(C \vee A')' \vee (A \vee C')'] \wedge (B \vee C')\}' | \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D' \wedge B)' \vee (D' \vee C)']\}$$

Aplicando las leyes de D'Morgan en los paréntesis que corresponda:

$$|[(A' \vee B') \wedge (B \vee A)] \vee \{[(C' \wedge A'') \vee (A' \wedge C'')] \wedge (B' \wedge C'')\}' | \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D'' \vee B') \vee (D'' \wedge C')]\}$$

Aplicando las idempotencias donde corresponde:

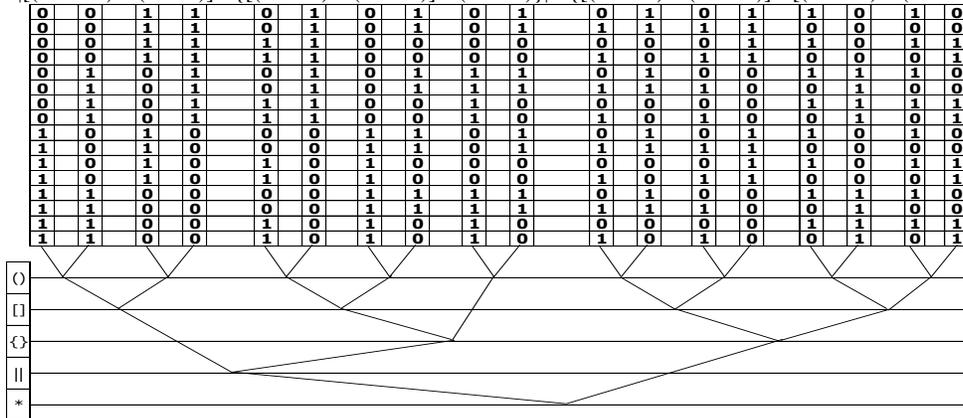
$$|[(A' \vee B') \wedge (B \vee A)] \vee \{[(C' \wedge A) \vee (A' \wedge C)] \wedge (B' \wedge C)\}' | \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D \vee B') \vee (D \wedge C')]\}$$

Se concluye que el polinomio transformado es:

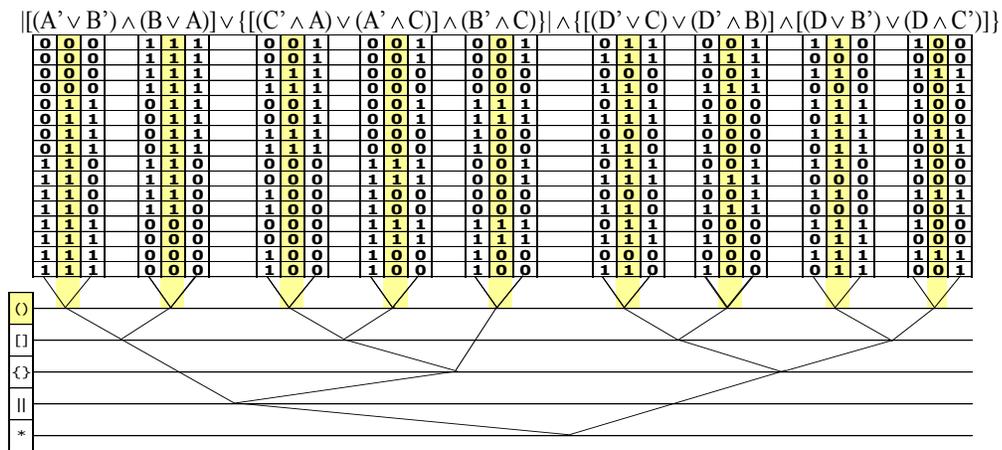
$$|[(A' \vee B') \wedge (B \vee A)] \vee \{[(C' \wedge A) \vee (A' \wedge C)] \wedge (B' \wedge C)\}' | \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D \vee B') \vee (D \wedge C')]\}$$

Tercera Etapa: Tabla lógica del polinomio transformado. Al igual que la tabla lógica del polinomio original, con fines didácticos, esta tabla se hará con la ayuda de su árbol de programación, como este polinomio tiene 4 variables, entonces su cantidad de filas es $2^4=16$, y el llenado de sus variables se hace usando el mismo criterio que se usó en la tabla de programación original:

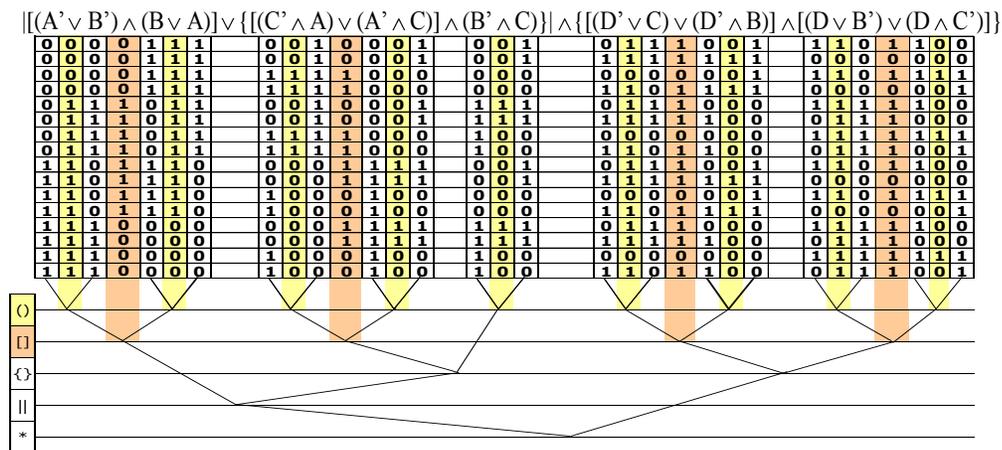
$$|[(A' \vee B') \wedge (B \vee A)] \vee \{[(C' \wedge A) \vee (A' \wedge C)] \wedge (B' \wedge C)\}' | \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D \vee B') \vee (D \wedge C')]\}$$



Resolviendo los paréntesis: En $(A \vee B)$, se escribe (0) en las casillas: 1, 2, 3 y 4; en el resto se escribe (1). En $(B \vee A)$, se escribe (0) en las casillas: 13, 14, 15 y 16; en el resto se escribe (1). En $(C \wedge A)$ se escribe (1) en las casillas: 3, 4, 7 y 8; en el resto se escribe (0). En $(A \wedge C)$ se escribe (1) en las casillas: 9, 10, 13 y 14; en el resto se escribe (0). En $(B \wedge C)$ se escribe (1) en las casillas: 5, 6, 13 y 14; en el resto se escribe (0). En $(D \vee C)$ se escribe (0) en las casillas: 3, 7, 11 y 15; en el resto se escribe (1). En $(D \wedge B)$ se escribe (1) en las casillas: 2, 4, 10 y 12; en el resto se escribe (0). En $(D \vee B')$ se escribe (0) en las casillas: 2, 4, 10 y 12; en el resto se escribe (1). En $(D \wedge C')$ se escribe (1) en las casillas: 3, 7, 11 y 15; en el resto se escribe (0):

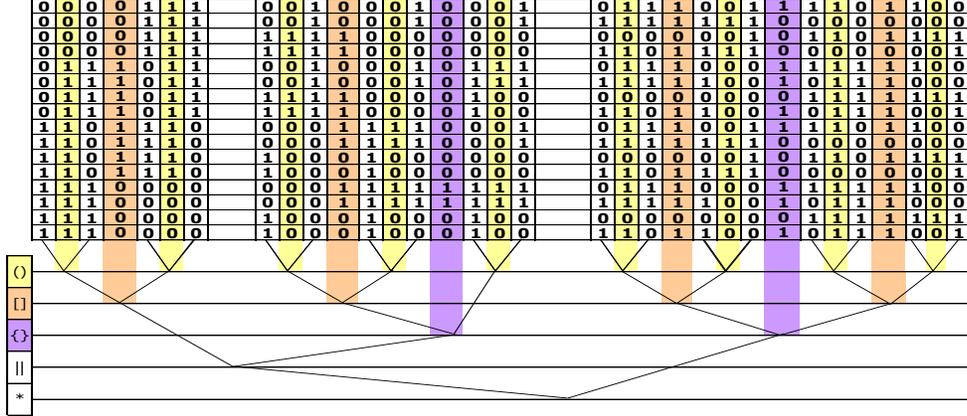


Resolviendo los corchetes: En $[(A \vee B') \wedge (B \vee A)]$, se escribe (1) en las casillas: 5, 6, 7, 8, 9, 10, 11 y 12; en el resto se escribe (0). En $\{[(C \wedge A) \vee (A' \wedge C)] \wedge (B' \wedge C)\}$, se escribe (0) en las casillas: 1, 2, 5, 6, 11, 12, 15 y 16; en el resto se escribe (1). En $\{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D \vee B') \vee (D \wedge C')]\}$, se escribe (0) en las casillas: 2, 4, 10, y 12; en el resto se escribe (1):



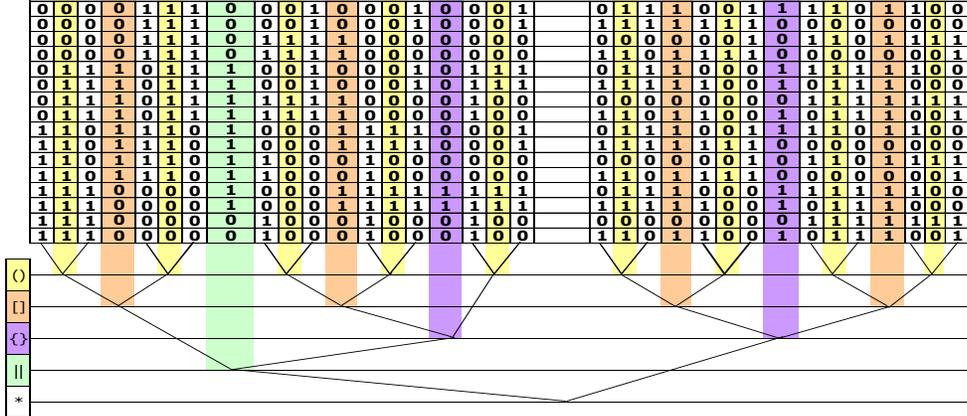
Resolviendo las llaves: En $\{[(C \wedge A) \vee (A' \wedge C)] \wedge (B' \wedge C)\}$, es una conjunción, se escribe (1) en las casillas: 13 y 14; en el resto se escribe (0). En $\{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D \vee B') \vee (D \wedge C')]\}$, es una conjunción, se escribe (1) en las casillas: 1, 5, 6, 8, 9, 13, 14, y 16; en el resto de las casillas se escribe (0). La tabla lógica queda:

$$[(A' \vee B') \wedge (B \vee A)] \vee \{[(C' \wedge A) \vee (A' \wedge C)] \wedge (B' \wedge C)\} \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D \vee B') \vee (D \wedge C')]\}$$



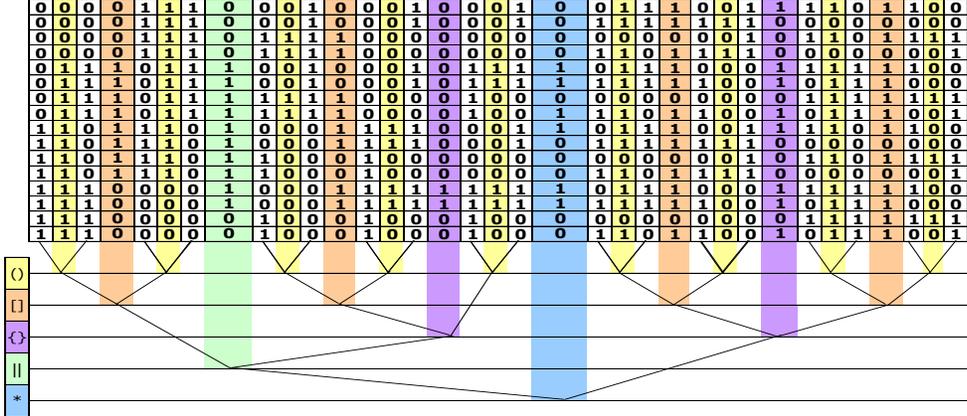
Resolviendo la barra: $[(A' \vee B') \wedge (B \vee A)] \vee \{[(C' \wedge A) \vee (A' \wedge C)] \wedge (B' \wedge C)\}$, es una disyunción, se escribe (0) en las casillas: 1, 2, 3, 4, 15 y 16; en el resto se escribe (1):

$$[(A' \vee B') \wedge (B \vee A)] \vee \{[(C' \wedge A) \vee (A' \wedge C)] \wedge (B' \wedge C)\} \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D \vee B') \vee (D \wedge C')]\}$$



Resolviendo el conectivo principal, por ser una conjunción, se escribe (1) en las casillas: 5, 6, 8, 9, 13 y 14; en el resto se escribe (0):

$$[(A' \vee B') \wedge (B \vee A)] \vee \{[(C' \wedge A) \vee (A' \wedge C)] \wedge (B' \wedge C)\} \wedge \{[(D' \vee C) \vee (D' \wedge B)] \wedge [(D \vee B') \vee (D \wedge C')]\}$$

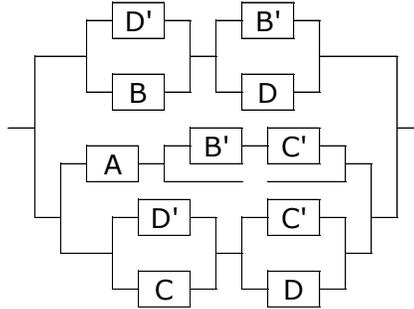


Se presenta a continuación la tabla final con las características que se han incluido dada su naturaleza de mostrar los estados del circuito:

de dichos circuitos. Para reforzar lo que recientemente se explicó, se deja al lector la verificación del siguiente ejercicio:

- Polinomio: $(D \leftrightarrow B) \vee \{ [A \rightarrow (B \vee C')] \wedge (D \leftrightarrow C) \}'$

- Circuito serie-paralelo correspondiente:



Operaciones del álgebra booleana y la miniaturización de circuitos series-paralelo:

Después de haber recorrido este camino, el lector avezado en los temas concernientes al álgebra de Boole naturalmente se puede preguntar ¿Es posible relacionar las operaciones del álgebra booleana con la miniaturización de circuitos serie-paralelo? La respuesta es sí, ya que las operaciones que son usadas en los polinomios lógicos no modifican su valor lógico o en nuestro caso su tabla lógica, pero a la hora de hacer los dibujos de esos circuitos se puede apreciar una reducción en su cantidad de componentes sin perder su funcionalidad. Para empezar, es necesario enlistar las leyes, operaciones y propiedades que se van a abordar y como con todos los fundamentos explicados en Las páginas anteriores se puede hacer las manipulaciones de rigor para aplicarlos en los circuitos series-paralelo:

$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$	(1) Distributividad de la conjunción respecto de la disyunción
$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$	(2) Distributividad de la disyunción respecto de la conjunción
$A \wedge B \equiv B \wedge A$	(3) Conmutatividad de la conjunción
$(A \wedge B)' \equiv A' \vee B'$	(4) Negación de la conjunción (D'Morgan)
$A \wedge A \equiv A$	(5) Idempotencia de la conjunción
$A \wedge A' \equiv 0$	(6) Conjunción de complementos (corto circuito)
$A \wedge 0 \equiv 0$	(7) Elemento absorbente de la conjunción
$A \wedge 1 \equiv A$	(8) Elemento neutro de la conjunción
$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$	(9) Asociatividad de la conjunción
$A \vee B \equiv B \vee A$	(10) Conmutatividad de la disyunción
$(A \vee B)' \equiv A' \wedge B'$	(11) Negación de la disyunción (D'Morgan)
$A \vee A \equiv A$	(12) Idempotencia de la disyunción
$A \vee A' \equiv 1$	(13) Disyunción de complementos (puente)

$A \vee 1 \equiv 1$	(14) Elemento absorbente de la disyunción
$A \vee 0 \equiv A$	(15) Elemento neutro de la disyunción
$(A \vee B) \vee C \equiv A \vee (B \vee C)$	(16) Asociatividad de la disyunción
$(A')' \equiv A$	(17) Idempotencia de la negación
$(0)' \equiv 1$	(18) Complemento del corto circuito
$(1)' \equiv 0$	(19) Complemento del puente
$A \rightarrow B \equiv A' \vee B$	(20) Ley del condicional
$(A \rightarrow B)' \equiv A \wedge B'$	(21) Negación del condicional
$A \rightarrow 0 \equiv A'$	(22) Condicional al corto circuito
$A \rightarrow 1 \equiv 1$	(23) Condicional al puente
$1 \rightarrow A \equiv A$	(24) Condicional del puente
$0 \rightarrow A \equiv 1$	(25) Condicional del corto circuito
$A \rightarrow A' \equiv A'$	(26) Condicional al complemento
$A' \rightarrow A \equiv A$	(27) Condicional del complemento
$A \leftrightarrow B \equiv (A' \vee B) \wedge (B' \vee A)$	(28) Ley del bicondicional
$(A \leftrightarrow B)' \equiv (A \wedge B') \vee (B \wedge A')$	(29) Negación del bicondicional
$A \leftrightarrow A' \equiv 0$	(30) Bicondicional de los complementos
$A \leftrightarrow B \equiv B \leftrightarrow A$	(31) Conmutatividad del bicondicional
$A \leftrightarrow 0 \equiv A'$	(32) Elemento inverso del bicondicional
$A \leftrightarrow 1 \equiv 1$	(33) Elemento absorbente del bicondicional

Ejemplo: Dado el siguiente polinomio $[(A \vee B) \wedge (C' \rightarrow A)] \vee \{ [A' \rightarrow (B' \rightarrow B)']' \vee (A \wedge D) \} \wedge (C' \leftrightarrow B)'$.

Determinar:

- 1.- Tabla lógica
- 2.- Polinomio equivalente con su respectiva tabla lógica y circuito.
- 3.- Polinomio equivalente reducido con su respectiva tabla lógica y circuito

Solución: Se trabajará en etapas como ya se ha trabajado en los anteriores ejercicios. En la primera etapa es simplemente hacer una tabla lógica sin modificar el polinomio original. En la segunda etapa, solamente se aplican al polinomio las leyes y propiedades enlistadas en la página anterior y el principio de esta página desde los signos de agrupación mayores hacia los menores para obtener el polinomio equivalente, luego se elabora su tabla lógica y se dibuja su circuito. Finalmente la tercera etapa consiste en aplicar las leyes y propiedades enlistadas en la página anterior y principio de ésta para reducir el polinomio, a continuación se elabora su tabla lógica y después se dibuja su circuito. Vale la pena aclarar que la única guía que se va a disponer para saber si no se ha cometido errores es la tabla de codificación del polinomio original, si la tabla de codificación del polinomio equivalente no coincide exactamente con ella entonces no se puede proceder a la tercera etapa, y la tercera etapa no estará completada de manera correcta si la tabla de codificación del polinomio equivalente reducido no coincide con las tablas de codificación anteriores.

Primera etapa: Elaboración de la tabla lógica del polinomio original...

$$[(A \vee B) \wedge (C' \rightarrow A)] \vee \{ [A' \rightarrow (B' \rightarrow B')] \}' \vee (A \wedge D) \} \wedge (C' \leftrightarrow B)'$$

1	1	1	0	1	1	0	0	0	0	1	1	0	1
1	1	1	0	1	1	0	0	0	0	1	0	0	1
1	1	1	1	1	1	0	0	0	1	1	1	1	1
1	1	1	1	1	1	0	0	0	1	0	1	1	1
1	0	0	1	1	0	1	1	1	1	1	0	0	0
1	0	0	1	1	0	1	1	1	1	0	0	0	0
1	0	1	1	1	0	1	1	1	1	1	1	1	0
1	0	1	1	1	0	1	1	1	1	0	1	1	0
0	1	0	0	0	1	0	0	0	0	1	0	0	1
0	1	0	0	0	1	0	0	0	0	0	0	0	1
0	1	1	1	0	1	0	0	0	0	1	1	1	1
0	1	1	1	0	1	0	0	0	0	0	1	1	1
0	0	0	0	0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0	0
0	0	1	0	0	1	1	1	1	0	1	1	1	0
0	0	1	0	0	1	1	1	1	0	0	1	1	0
0	0	1	0	0	1	1	1	1	0	1	1	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0	0
0	0	1	1	0	1	1	1	1	0	1	1	1	0
0	0	1	1	0	1	1	1	1	0	0	1	1	0

∅
□
⊆
∥
*

Se procederá a rellenar esta tabla siguiendo este resumen de criterios:

<p>Para la conjunción (∧):</p> <p>Busque las casillas donde ambos elementos sean (1) y escriba (1); en el resto de casillas escriba (0). Si está negada, haga la misma búsqueda pero en vez de escribir (1), escriba (0); en el resto de casillas escriba (1)</p>	<p>Para la disyunción (∨):</p> <p>Busque las casillas donde ambos elementos sean (0) y escriba (0); en el resto de casillas escriba (1). Si está negada, haga la misma búsqueda pero en vez de escribir (0), escriba (1); en el resto de casillas escriba (0)</p>
<p>Para el condicional (→):</p> <p>Busque las casillas donde el elemento de la izquierda sea (1) y el de la derecha (0) y escriba (0); en el resto de casillas escriba (1). Si está negado, haga la misma búsqueda pero en vez de escribir (0), escriba (1); en el resto de casillas escriba (0)</p>	<p>Para el bicondicional (↔):</p> <p>Busque las casillas donde ambos elementos sean iguales (ambos=1 o ambos=0) y escriba (1); en el resto de casillas escriba (0). Si está negado, haga la misma búsqueda pero en vez de escribir (1), escriba (0); en el resto de casillas escriba (1)</p>

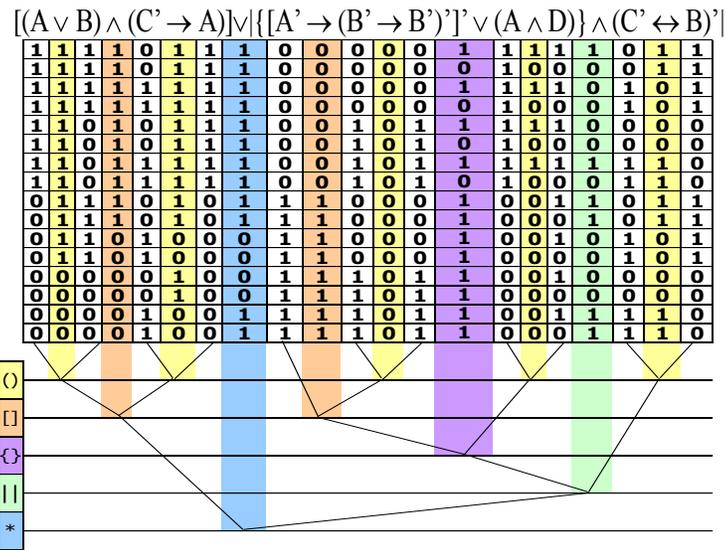
Resolviendo los paréntesis:

$$[(A \vee B) \wedge (C' \rightarrow A)] \vee \{ [A' \rightarrow (B' \rightarrow B')] \}' \vee (A \wedge D) \} \wedge (C' \leftrightarrow B)'$$

1	1	1	0	1	1	0	0	0	0	1	1	0	1	1	
1	1	1	0	1	1	0	0	0	0	1	0	0	0	1	1
1	1	1	1	1	1	0	0	0	0	1	1	1	1	0	1
1	1	1	1	1	1	0	0	0	0	1	0	0	1	0	1
1	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0
1	1	0	0	1	1	0	1	0	1	1	0	0	0	0	0
1	1	0	1	1	1	0	1	0	1	1	1	1	1	1	0
1	1	0	1	1	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	1	0	1	0	0	0	0	0	1	0	1	1
0	1	1	0	1	0	1	0	0	0	0	0	0	0	1	1
0	1	1	1	0	0	1	0	0	0	0	0	1	0	1	1
0	1	1	1	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	0	1	0	1	1	0	1	0	0	1	0	0	0
0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0
0	0	0	1	0	0	1	1	1	0	1	0	0	1	1	0
0	0	0	1	0	0	1	1	1	0	1	0	0	1	1	0
0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	0

∅
□
⊆
∥
*

Resolviendo el conectivo principal:



La tabla lógica con los estados del polinomio es la siguiente:

$$[(A \vee B) \wedge (C' \rightarrow A)] \vee \{ [A' \rightarrow (B' \rightarrow B')] \vee (A \wedge D) \} \wedge (C' \leftrightarrow B)'$$

1	1	1	1	0	1	1	1	0	0	0	0	0	1	1	1	1	0	1	1	ENCENDIDO	
1	1	1	1	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	1	ENCENDIDO
1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	1	0	1	ENCENDIDO	
1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	1	0	1	ENCENDIDO	
1	1	0	1	0	1	1	1	0	0	1	0	1	1	1	1	0	0	0	0	ENCENDIDO	
1	1	0	1	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	0	ENCENDIDO	
1	1	0	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	0	ENCENDIDO	
1	1	0	1	1	1	1	1	0	0	1	0	1	0	1	0	0	1	1	0	ENCENDIDO	
0	1	1	1	0	1	0	1	1	1	0	0	0	1	0	0	1	1	0	1	ENCENDIDO	
0	1	1	1	0	1	0	1	1	1	0	0	0	1	0	0	1	0	1	1	ENCENDIDO	
0	1	1	0	1	0	0	0	1	1	0	0	0	1	0	0	1	0	1	0	ENCENDIDO	
0	1	1	0	1	0	0	0	1	1	0	0	0	1	0	0	0	1	0	1	APAGADO	
0	0	0	0	0	1	0	0	1	1	1	0	1	1	0	0	1	0	0	0	APAGADO	
0	0	0	0	0	1	0	0	1	1	1	0	1	1	0	0	0	0	0	0	APAGADO	
0	0	0	0	1	0	0	1	1	1	1	0	1	1	0	0	1	1	1	0	ENCENDIDO	
0	0	0	0	1	0	0	1	1	1	1	0	1	1	0	0	0	1	1	0	ENCENDIDO	

La Tabla de codificación del polinomio original es:

1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Segunda Etapa:

Elaboración del polinomio equivalente partiendo del polinomio original...

Polinomio Original:

$$[(A \vee B) \wedge (C' \rightarrow A)] \vee \{ [A' \rightarrow (B' \rightarrow B')] \vee (A \wedge D) \} \wedge (C' \leftrightarrow B)'$$

Aplicando (21) en $[A' \rightarrow (B' \rightarrow B')]'$:

$$[(A \vee B) \wedge (C' \rightarrow A)] \vee \{ [A' \wedge (B' \rightarrow B')] \vee (A \wedge D) \} \wedge (C' \leftrightarrow B)'$$

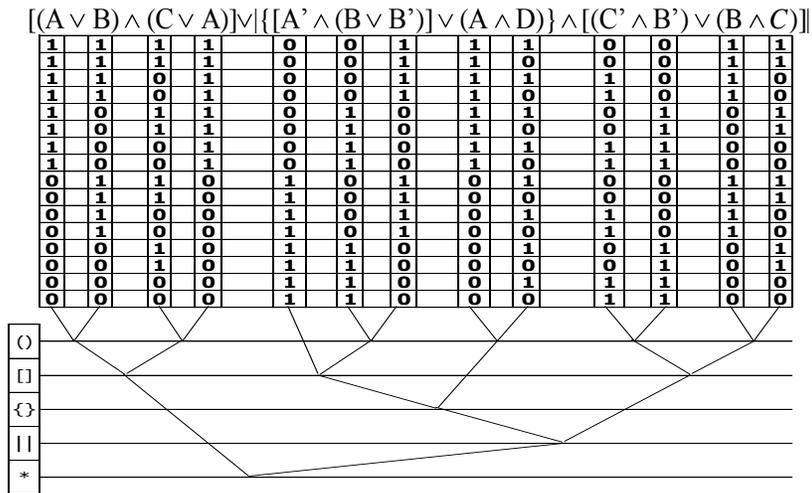
Aplicando (20) en $(C' \rightarrow A)$, $(B' \rightarrow B')$ y (29) en $(C' \leftrightarrow B)'$:

$$[(A \vee B) \wedge (C \vee A)] \vee \{ [A' \wedge (B \vee B')] \vee (A \wedge D) \} \wedge [(C' \wedge B') \vee (B \wedge C)]$$

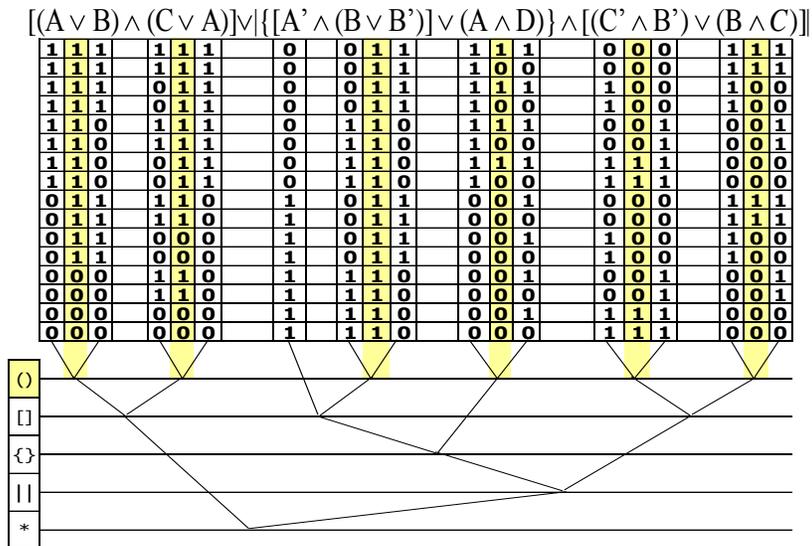
Se concluye que el Polinomio Equivalente es:

$$[(A \vee B) \wedge (C \vee A)] \vee \{ [A' \wedge (B \vee B')] \vee (A \wedge D) \} \wedge [(C' \wedge B') \vee (B \wedge C)]$$

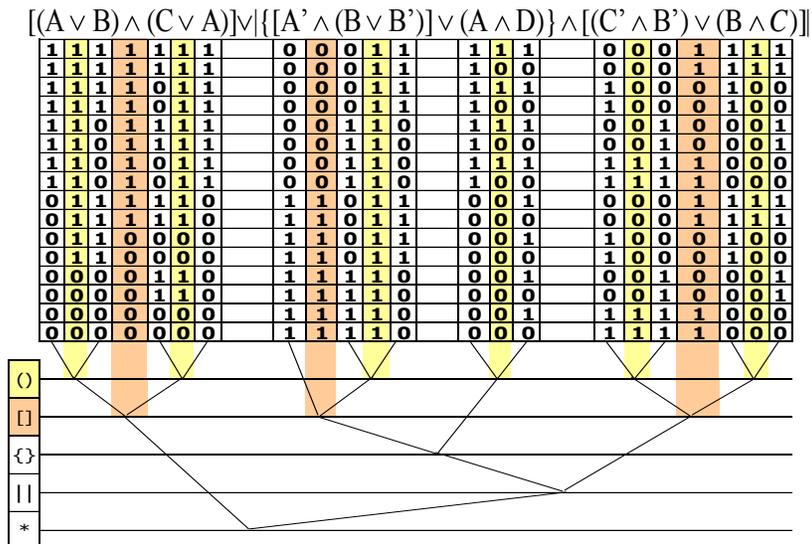
Elaboración de la tabla lógica del Polinomio Equivalente...



Resolviendo los paréntesis:



Resolviendo los corchetes:



La tabla lógica con los estados del polinomio es la siguiente:

$$[(A \vee B) \wedge (C \vee A)] \vee \{ [A' \wedge (B \vee B')] \vee (A \wedge D) \} \wedge [(C' \wedge B') \vee (B \wedge C)]$$

1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	1	1	1	1	ENCENDIDO
1	1	1	1	1	1	1	0	0	0	1	1	0	1	0	0	0	0	0	1	1	1	1	ENCENDIDO
1	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1	0	1	0	0	0	1	0	ENCENDIDO
1	1	1	1	0	1	1	1	0	0	0	1	1	0	1	0	0	1	0	0	0	1	0	ENCENDIDO
1	1	0	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0	0	1	0	0	0	ENCENDIDO
1	1	0	1	1	1	1	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	ENCENDIDO
1	1	0	1	0	1	1	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	0	ENCENDIDO
1	1	0	1	0	1	1	1	0	0	1	1	0	0	1	0	0	0	1	1	1	1	0	ENCENDIDO
0	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	1	1	0	0	1	1	1	ENCENDIDO
0	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	0	1	0	0	0	1	1	ENCENDIDO
0	1	1	0	0	0	0	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	1	APAGADO
0	1	1	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	1	APAGADO
0	0	0	0	1	1	0	0	1	1	1	0	1	0	0	1	0	0	0	1	0	0	0	APAGADO
0	0	0	0	1	1	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	ENCENDIDO
0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	1	1	1	1	1	1	1	0	ENCENDIDO
0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	1	1	1	1	1	1	0	ENCENDIDO

La Tabla de codificación del Polinomio Equivalente es:

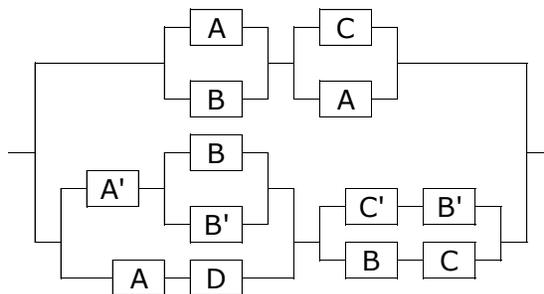
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Verificando las tablas de codificación, se tiene:

Tabla Original:	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
Tabla Equivalente:	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
Verificación:	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Como se puede apreciar, Ambas tablas lógicas son EQUIVALENTES.

Circuito serie-paralelo equivalente:



Tercera Etapa:

Elaboración del Polinomio Equivalente Reducido partiendo del Polinomio Equivalente...

Polinomio Equivalente:

$$[(A \vee B) \wedge (C \vee A)] \vee \{ [A' \wedge (B \vee B')] \vee (A \wedge D) \} \wedge [(C' \wedge B') \vee (B \wedge C)]$$

Aplicando (10) en $(C \vee A)$, (13) en $(B \vee B')$:

$$[(A \vee B) \wedge (A \vee C)] \vee \{ [A' \wedge 1] \vee (A \wedge D) \} \wedge [(C' \wedge B') \vee (B \wedge C)]$$

Aplicando (8) en $(A' \wedge 1)$, (2) en $[(A \vee B) \wedge (A \vee C)]$:

$$[A \vee (B \wedge C)] \vee \{ [A' \vee (A \wedge D)] \wedge [(C' \wedge B') \vee (B \wedge C)] \}$$

Aplicando (2) en $[A' \vee (A \wedge D)]$:

$$[A \vee (B \wedge C)] \vee \{ [(A' \vee A) \wedge (A' \vee D)] \wedge [(C' \wedge B') \vee (B \wedge C)] \}$$

Aplicando (13) en $(A' \vee A)$:

$$[A \vee (B \wedge C)] \vee \{ [1 \wedge (A' \vee D)] \wedge [(C' \wedge B') \vee (B \wedge C)] \}$$

Aplicando (8) en $[1 \wedge (A' \vee D)]$:
 $[A \vee (B \wedge C)] \vee \{(A' \vee D) \wedge [(C' \wedge B') \vee (B \wedge C)]\}$

Aplicando (2) en $[A \vee (B \wedge C)] \vee \{(A' \vee D) \wedge [(C' \wedge B') \vee (B \wedge C)]\}$
 $\{[A \vee (B \wedge C)] \vee (A' \vee D)\} \wedge \{[A \vee (B \wedge C)] \vee [(C' \wedge B') \vee (B \wedge C)]\}$

Aplicando (10) en $[(C' \wedge B') \vee (B \wedge C)]$:
 $\{[A \vee (B \wedge C)] \vee (A' \vee D)\} \wedge \{[A \vee (B \wedge C)] \vee [(B \wedge C) \vee (C' \wedge B')]\}$

Aplicando (16) en $[A \vee (B \wedge C)] \vee [(B \wedge C) \vee (C' \wedge B')]$
 $\{[A \vee (B \wedge C)] \vee (A' \vee D)\} \wedge \{[A \vee (C' \wedge B')] \vee [(B \wedge C) \vee (B \wedge C)]\}$

Aplicando (12) en $[(B \wedge C) \vee (B \wedge C)]$:
 $\{[A \vee (B \wedge C)] \vee (A' \vee D)\} \wedge \{[A \vee (C' \wedge B')] \vee (B \wedge C)\}$

Aplicando (10) y (16) en $[A \vee (C' \wedge B')] \vee (B \wedge C)$
 $\{[A \vee (B \wedge C)] \vee (A' \vee D)\} \wedge \{[A \vee (B \wedge C)] \vee (C' \wedge B')\}$

Aplicando (2) en $\{[A \vee (B \wedge C)] \vee (A' \vee D)\} \wedge \{[A \vee (B \wedge C)] \vee (C' \wedge B')\}$
 $[A \vee (B \wedge C)] \vee [(A' \vee D) \wedge (C' \wedge B')]$

Se concluye que el Polinomio Equivalente Reducido es:
 $[A \vee (B \wedge C)] \vee [(A' \vee D) \wedge (C' \wedge B')]$

Elaboración de la tabla lógica correspondiente al Polinomio Equivalente Reducido:

$[A \vee (B \wedge C)] \vee [(A' \vee D) \wedge (C' \wedge B')]$

1	1	1	0	1	0	0
1	1	1	0	0	0	0
1	1	0	0	1	1	0
1	1	0	0	0	0	1
1	0	1	0	1	0	1
1	0	1	0	0	0	1
1	0	0	0	1	1	1
1	0	0	0	0	0	1
0	1	1	1	1	0	0
0	1	1	1	1	1	0
0	1	0	1	1	0	0
0	0	1	1	1	1	0
0	0	1	1	1	0	1
0	0	1	1	0	0	1
0	0	0	1	1	1	1
0	0	0	1	0	1	1

()
[]
*

Resolviendo los paréntesis:

$[A \vee (B \wedge C)] \vee [(A' \vee D) \wedge (C' \wedge B')]$

1	1	1	1	0	1	1	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	0	0	0	1	1	1	0	0
1	1	0	0	0	0	0	1	0	0
1	0	0	1	0	1	1	0	0	1
1	0	0	1	0	0	0	0	0	1
1	0	0	0	0	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1
0	1	1	1	1	1	1	0	0	0
0	1	1	1	1	1	0	0	0	0
0	1	0	0	1	1	1	1	0	0
0	1	0	0	1	1	0	1	0	0
0	0	0	1	1	1	1	0	0	1
0	0	0	1	1	1	0	0	0	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	0	1	1	1

()
[]
*

Resolviendo los corchetes

$[A \vee (B \wedge C)] \vee [(A' \vee D) \wedge (C' \wedge B')]$

1	1	1	1	1	0	1	1	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	1	1	0	1	0	0
1	1	1	0	0	0	0	0	0	1	0	0
1	1	0	0	1	0	1	1	0	0	0	1
1	1	0	0	1	0	0	0	0	0	0	1
1	1	0	0	0	0	1	1	1	1	1	1
1	1	0	0	0	0	1	1	1	0	0	1
0	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0	0
0	1	0	0	0	1	1	1	1	0	1	0
0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	0	1	1	1	1	0	0	0	1
0	0	0	0	1	1	1	0	0	0	0	1
0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	1	1	1	1

()
[]
*

Resolviendo el conectivo principal:

$$[A \vee (B \wedge C)] \vee [(A' \vee D) \wedge (C' \wedge B')]$$

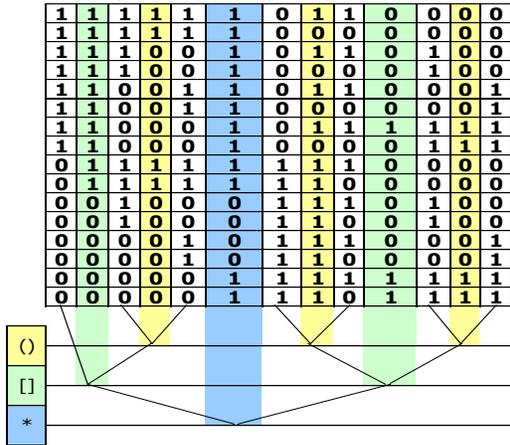


Tabla lógica con los estados incluidos:

$$[A \vee (B \wedge C)] \vee [(A' \vee D) \wedge (C' \wedge B')]$$

1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	ENCENDIDO
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	ENCENDIDO
1	1	1	0	0	1	0	1	1	0	1	0	0	0	0	0	ENCENDIDO
1	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	ENCENDIDO
1	1	0	0	1	1	0	1	1	0	0	0	0	0	0	1	ENCENDIDO
1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	ENCENDIDO
1	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	ENCENDIDO
1	1	0	0	1	1	0	0	0	0	0	1	1	1	1	1	ENCENDIDO
0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	ENCENDIDO
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	ENCENDIDO
0	0	1	0	0	0	1	1	1	0	1	0	1	0	0	0	ENCENDIDO
0	0	1	0	0	0	1	1	0	0	1	0	0	1	0	0	ENCENDIDO
0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	1	ENCENDIDO
0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	ENCENDIDO
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	ENCENDIDO
0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	ENCENDIDO
0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	ENCENDIDO
0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	ENCENDIDO
0	0	1	0	0	0	0	1	1	1	0	1	0	0	0	0	APAGADO
0	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	APAGADO
0	0	0	0	1	0	1	1	1	0	0	0	0	0	1	0	APAGADO
0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	APAGADO
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	ENCENDIDO
0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	ENCENDIDO

La Tabla de codificación del Polinomio Equivalente Reducido es:

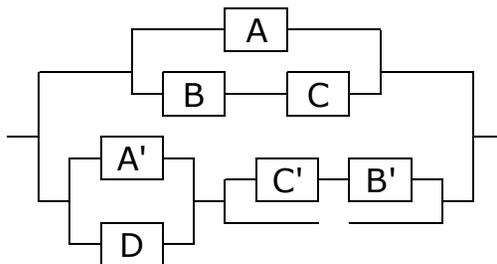
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Verificando las tablas de codificación, se tiene:

Tabla Original:	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
Tabla Equivalente:	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
Tabla Reducida:	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
Verificación:	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Como se puede apreciar, TODAS las tablas lógicas son EQUIVALENTES.

Circuito serie-paralelo equivalente:

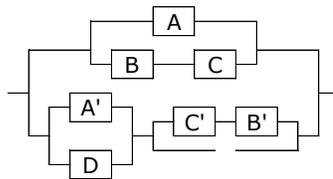


Consideración adicional acerca de los circuitos que resultan de los polinomios equivalentes y reducidos:

Anteriormente se habló de las "líneas incompletas" que aparecen en algunos dibujos, pues es momento de profundizar en este tema para cubrir de manera elemental los aspectos de esta aplicación del álgebra de Boole en los circuitos serie-paralelo. Como ya se explicó, las líneas incompletas aparecen en los dibujos donde la agrupación de las variables no siguen la regla: "las series se agrupan de dos en dos, de izquierda a derecha y los paralelos se agrupan de dos en dos, de arriba hacia abajo". Pues bien, en realidad las líneas incompletas solamente aparecen en

las agrupaciones en serie, los paralelos pueden dibujarse sin ningún problema independientemente de su modo de agrupación. Entonces aquí surge una cuestión que a lo mejor es irrelevante desde el punto de vista matemático, ya que lo que importa es la tabla lógica y ésta es equivalente a la tabla del polinomio lógico original; pero tal vez si tenga relevancia si se pretende construir estos circuitos ya que estas líneas incompletas van a representar pedazos de conductores que se van a desperdiciar y eso dista mucho del uso eficiente de los recursos. Como el problema se centra en las series entonces, podemos auxiliarnos en una propiedad de las conjunciones que en el presente inciso se encuentra identificado como: (9) Asociatividad de la conjunción: $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$. Se propone usar esta propiedad para aquellas series donde se presenten este problema de las líneas incompletas. A continuación se aplicará esto en el resultado del circuito de la página anterior.

Ejemplo: Use la Asociatividad para eliminar la línea incompleta en el siguiente circuito:



Solución: Se traduce el dibujo del circuito lógico en su respectivo polinomio, se hace su tabla lógica y se obtiene su tabla de codificación (la cual va a ser la guía para saber si vamos bien). Acto seguido se aplica la asociatividad en la parte que corresponde a la línea incompleta, luego se elabora su tabla lógica y se obtiene su tabla de codificación, si coincide con la tabla de codificación del circuito original significa que son equivalentes y se pueden usar con seguridad.

Traducción: $[A \vee (B \wedge C)] \vee [(A' \vee D) \wedge (C' \wedge B')]$

Tabla lógica:

$[A \vee (B \wedge C)] \vee [(A' \vee D) \wedge (C' \wedge B')]$													
1	1	1	1	1	1	0	1	1	0	0	0	0	ENCENDIDO
1	1	1	1	1	1	0	0	0	0	0	0	0	ENCENDIDO
1	1	1	0	0	1	0	1	1	0	1	0	0	ENCENDIDO
1	1	1	0	0	1	1	0	0	0	0	1	0	ENCENDIDO
1	1	0	0	1	1	0	0	0	0	0	0	1	ENCENDIDO
1	1	0	0	0	1	0	1	1	1	1	1	1	ENCENDIDO
1	1	0	0	0	1	0	0	0	0	1	1	1	ENCENDIDO
0	1	1	1	1	1	1	1	1	0	0	0	0	ENCENDIDO
0	1	1	1	1	1	1	1	0	0	0	0	0	ENCENDIDO
0	0	1	0	0	0	1	1	1	0	1	0	0	APAGADO
0	0	1	0	0	0	1	1	0	0	1	0	0	APAGADO
0	0	0	0	1	0	1	1	1	0	0	0	1	APAGADO
0	0	0	0	1	0	1	1	0	0	0	0	1	APAGADO
0	0	0	0	0	1	1	1	1	1	1	1	1	ENCENDIDO
0	0	0	0	0	1	1	1	0	1	1	1	1	ENCENDIDO

Tabla de codificación del polinomio original:

1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Aplicando la ley asociativa y reorganizando el polinomio bajo el criterio "las series se organizan de dos en dos y de izquierda a derecha":

Polinomio Original:

$$[A \vee (B \wedge C)] \vee [(A' \vee D) \wedge (C' \wedge B')]$$

Aplicando (9) en: $[(A' \vee D) \wedge (C' \wedge B')]$

$$[A \vee (B \wedge C)] \vee \{[(A' \vee D) \wedge C'] \wedge B'\}$$

Se concluye que el polinomio lógico equivalente es: $[A \vee (B \wedge C)] \vee \{[(A' \vee D) \wedge C'] \wedge B'\}$

Tabla lógica del polinomio equivalente:

$$[A \vee (B \wedge C)] \vee \{[(A' \vee D) \wedge C'] \wedge B'\}$$

1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	ENCENDIDO
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	ENCENDIDO
1	1	1	0	0	1	0	1	1	1	1	0	0	0	0	0	ENCENDIDO
1	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	ENCENDIDO
1	1	0	0	1	1	0	1	1	0	0	0	0	0	1	0	ENCENDIDO
1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	ENCENDIDO
1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	ENCENDIDO
1	1	0	0	0	1	0	0	0	0	0	1	0	1	0	1	ENCENDIDO
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	ENCENDIDO
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	ENCENDIDO
0	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	APAGADO
0	0	1	0	0	0	1	1	0	1	1	1	0	0	0	0	APAGADO
0	0	0	0	1	0	1	1	1	0	0	0	0	1	0	0	APAGADO
0	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	APAGADO
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	ENCENDIDO
0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	ENCENDIDO

Tabla de codificación del polinomio equivalente:

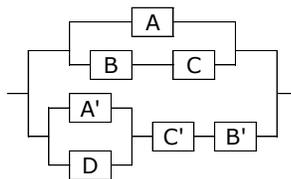
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Verificando las tablas de codificación, se tiene:

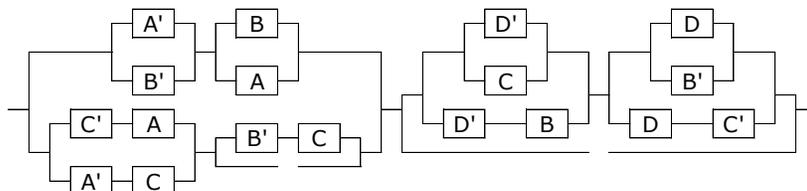
Tabla Original:	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
Tabla Equivalente:	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
Verificación:	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Como se puede apreciar, las tablas lógicas son EQUIVALENTES.

Circuito serie-paralelo equivalente:



Se deja al lector la verificación del polinomio lógico equivalente al siguiente circuito:



Polinomio equivalente:

$$\{[(A' \vee B') \wedge (B \vee A)] \vee \{[(C' \wedge A) \vee (A' \wedge C)] \wedge B'\} \wedge C\} \wedge [(D' \vee C) \vee (D' \wedge B)] \vee \{[(D \vee B') \vee (D \wedge C')]\}$$

Conclusiones:

Después de exponer esta aplicación de las leyes lógicas en los circuitos serie-paralelo, se evidencia la afirmación de Russell y Wittgenstein, que la estructura del mundo es la de la lógica matemática, ya que en el mundo real se cumplen leyes que están regidas por las leyes matemáticas. Es más si deseamos hablar del azar, ya existen modelos matemáticos que están encaminados a "domar esa bestia", pero son en realidad primeros pasos.

También se afirma el punto de vista del positivismo lógico, el cual propone que para conformar el conocimiento se deben hacer mediante las propuestas analíticas y las sintéticas. Las analíticas cuya verdad o falsedad se obtiene por medio de la lógica y las sintéticas que son las que requieren alguna forma de verificación además de la lógica. Pues bien, en este artículo los polinomios lógicos tratados fueron exhaustivamente verificados mediante leyes lógicas y los circuitos serie-paralelo subyacentes pueden ser verificados mediante construcciones empíricas usando por ejemplo materiales de provecho. Esto a despecho de lo que afirma Russell, que nuestra experiencia directa es lo más importante en la adquisición del conocimiento y todo lo demás puede solamente ser inferido o razonado pero no conocidos directamente, claro esto ocurre con mucha frecuencia en la matemática porque se trabajan con entes abstractos: números, símbolos, operaciones, etc. Y en realidad estos objetos matemáticos no se pueden palpar mediante algún sentido como el tacto, el olfato, etc. Pero la corriente eléctrica si se puede percibir y sentir sus escalofriantes y mortales efectos apenas entramos en contacto con ella.

Se recomienda, darle a este ejercicio académico su correspondiente difusión ya que estos ejercicios académicos son la base para futuros adelantos, a través de la Educación y en estos tiempos donde la tecnología nos invade cada espacio es mejor irse adaptando de una buena vez.

Referencias Bibliográficas:

Afcha, Karim (1991). Temas Fundamentales de Lógica Matemática. Universidad de Carabobo. Valencia - Venezuela.

Brown, Frank (2003). Razonamiento Booleano: La lógica de las ecuaciones booleanas (2ª edición). Nueva York: Dover Publications.

Sáenz, Jorge; Gil, Fanny; López, Belkis; Romero, Neptalí; Bethelmy, José (2006). Fundamentos de la matemática. Editorial Hipotenusa. Barquisimeto - Venezuela.

Russell, B. (1956): Logic and Knowledge. Essays 1901-1950. Londres, Georges Allen and Unwin Ltd. Traducción castellana: Lógica y Conocimiento. Madrid, Taurus, 1981

Wittgenstein, L. (1921): Tractatus lógico-philosophicus. Traducción, introducción y notas de Luis M. Valdés Villanueva, Madrid, Tecnos, 2002.