

3  
3oj.



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
" ARAGON "

SIMPLIFICACION Y GRAFICACION  
DE CIRCUITOS LOGICOS

TESIS CON  
FALLA DE C...

**T E S I S**

QUE PARA OBTENER EL TITULO DE  
INGENIERO EN COMPUTACION  
P R E S E N T A :  
CUELLAR DIAZ ELIZABETH

DIRECTOR DE TESIS:  
ING. VICTOR HUGO SALAZAR NOGALES

MEXICO, D. F.

FEBRERO DE 1992



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INTRODUCCION .....	111
<b>I. INTRODUCCION A FUNCIONES DE BOOLE .....</b>	<b>1</b>
<b>II. PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>13</b>
II.1. Definición del problema .....	13
II.2. Alternativas de solución .....	14
II.2.1. Algebra de Boole .....	14
II.2.2. Mapa de Karnaugh .....	19
II.2.3. Quine McCluskey .....	33
II.3. Justificación de la alternativa a seleccionar .....	48
<b>III. ANALISIS DEL SISTEMA .....</b>	<b>49</b>
III.1. Definición del sistema .....	49
III.2. Requerimientos del sistema .....	50
<b>IV. DISEÑO DEL SISTEMA .....</b>	<b>58</b>
IV.1. Organización de los datos .....	58
IV.2. Diseño arquitectónico .....	61
IV.3. Diseño detallado .....	76
<b>V. DESARROLLO DEL SISTEMA .....</b>	<b>79</b>
V.1. Elección del lenguaje .....	79
V.2. Codificación .....	80
<b>VI. PRUEBAS Y MANUAL DE USUARIO .....</b>	<b>85</b>
VI.1. Pruebas .....	85
VI.2. Manual de usuario .....	86
<b>CONCLUSIONES .....</b>	<b>97</b>

<b>REFERENCIAS</b>	.....	<b>99</b>
<b>BIBLIOGRAFIA</b>	.....	<b>100</b>
<b>INDICE DE TABLAS Y FIGURAS</b>	.....	<b>101</b>

En el diseño de sistemas digitales se utilizan circuitos lógicos que son definidos por funciones de Boole. En su realización física es conveniente que se logre el menor costo posible. Para ello, se cuenta con métodos de minimización de funciones que serán descritos posteriormente.

La finalidad de este trabajo es presentar un sistema automatizado que minimize las funciones y grafique el diagrama lógico que describe el comportamiento del circuito lógico.

Para desarrollar el "Sistema de Minimización", existe la necesidad de emplear un enfoque sistemático, por lo que se utiliza el método tradicional de Ingeniería de Programación, que es el modelo de fases del ciclo de vida. Las fases que conforman el ciclo de vida son: Análisis, Diseño, Instrumentación, Prueba y Mantenimiento (FA187).

Esta tesis es dividida en seis capítulos. El capítulo I expone un panorama general de las funciones de Boole, así como su representación y clasificación, se analizan los elementos que conforman un sistema digital y su clasificación. El capítulo II plantea el porque del sistema y su justificación, se analizan cada una de las alternativas posibles para resolver el problema planteado y bajo ciertos criterios se selecciona el método desarrollado en computadora. En el capítulo III, por medio de un diagrama de flujo de datos, se representan los datos y procesos que el sistema utiliza y realiza respectivamente para cumplir con un objetivo, se especifican todas las funciones a realizar, se crea un diccionario de datos con el conjunto de datos a utilizar y una tabla de excepciones con mensajes de aviso o de error. En el capítulo IV se diseñan las estructuras de datos y por medio de pseudocódigo estructurado se detallan los módulos que intervienen en el sistema. La elección del lenguaje a utilizar y la traducción de los pseudocódigos a una estructura de programa se realiza en el capítulo V. En el capítulo VI se revisan y validan cada uno de los resultados esperados, efectuándose una serie de pruebas que permitan detectar la mayor cantidad de errores en el sistema tanto lógicos como funcionales.

**INTRODUCCION A FUNCIONES DE BOOLE**

La necesidad de construir circuitos lógicos al menor costo posible dió origen a métodos que minimicen funciones lógicas. El "Sistema de Minimización" posee la metodología automatizada que cubre esa necesidad.

Debido a que el objetivo de esta tesis es desarrollar un sistema que sirva como una herramienta en el área digital, es necesario definir y plantear todos los elementos de un Sistema Digital.

Un Sistema es un grupo de partes integradas que tienen el propósito común de lograr algún o algunos objetivos (SAND85).

Un Sistema Digital es cualquier sistema que transmite o procesa información, la cual es representada por cantidades físicas llamadas señales, que toman valores discretos.

El diseño de sistemas digitales está formado de tres módulos que en un momento determinado se interrelacionan para lograr un objetivo común.

- 1) Diseño del sistema
- 2) Diseño lógico
- 3) Diseño del circuito

El diseño del sistema corresponde a la división del problema en pequeños bloques llamados subsistemas que poseen características propias.

El diseño lógico es utilizado para interconectar bloques lógicos que en conjunto realizan una función en especial.

El diseño de circuitos es utilizado para la interconexión de componentes especiales como resistores, diodos y transistores, entre otros.

Cada uno de estos diseños deben ser estudiados por separado para tener una visión general de todo el sistema digital, aquí se analizará el módulo de diseño lógico. En la figura 1.1 se muestra el diagrama general de un sistema digital.

La información que es transmitida en un sistema digital fluye a través de circuitos lógicos como señales binarias. Los circuitos lógicos llamados compuertas son bloques de circuitería que cuentan con una salida y varias entradas. Las señales de entrada y salida son voltajes que representan al "1" y al "0" lógicos.

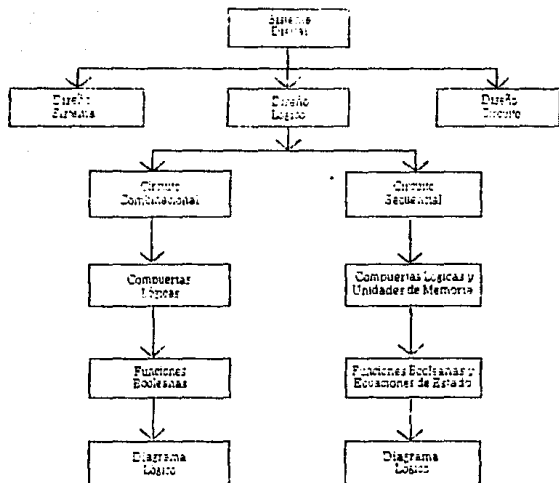


Figura 1.1. Diagrama General de un Sistema Digital

La función lógica de una compuerta es representada por una función de Boole. Su representación gráfica se muestra en la tabla 1.1, en donde también se observa la tabla de verdad, función de Boole y operador de cada compuerta (MOR33).

#### Funciones de Boole.

Una función de Boole de  $n$  terminales de entrada  $X_1, X_2, X_3, \dots, X_n$  es definida por  $F(X_1, X_2, X_3, \dots, X_n)$  y es una descripción matemática del comportamiento de un circuito lógico. Los símbolos  $X_1, X_2, X_3, \dots, X_n$  representan variables que pueden tomar valores de 0 a 1.









OPERADOR	SÍMBOLO	FUNCION DE BOOLE	TABLA DE VERDAD															
AND		$F = AB$	<table border="1"> <tr><td>A</td><td>B</td><td>F</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = A + B$	<table border="1"> <tr><td>A</td><td>B</td><td>F</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
INVERSOR		$F = \bar{A}$	<table border="1"> <tr><td>A</td><td>F</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	F	0	1	1	0									
A	F																	
0	1																	
1	0																	
BUFFER		$F = A$	<table border="1"> <tr><td>A</td><td>F</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table>	A	F	0	0	1	1									
A	F																	
0	0																	
1	1																	
NAND		$F = (A + B)'$	<table border="1"> <tr><td>A</td><td>B</td><td>F</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = (AB)'$	<table border="1"> <tr><td>A</td><td>B</td><td>F</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
OR EXCLUSIVA		$F = A'B + AB'$	<table border="1"> <tr><td>A</td><td>B</td><td>F</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
NOR EXCLUSIVA		$F = A'B' + AB$	<table border="1"> <tr><td>A</td><td>B</td><td>F</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Tabla 1.1. Compuertas lógicas



## 1. INTRODUCCION A FUNCIONES DE BOOLE

La razón de utilizar un modelo matemático es que puede ser analizado de varias formas y hacer posible varios diseños para elegir el más conveniente.

Cada una de las salidas del sistema está formada por un conjunto de expresiones de Boole. La expresión más simple está formada por una constante o una variable, como 0, A, D'. Expresiones más complicadas son formadas con combinaciones de operaciones de AND y OR denominándolos términos producto y términos suma.

$$\begin{aligned} &AB + AC \\ &AB' + C \\ &A(CD + E)' + AD \end{aligned}$$

Toda función de Boole se puede representar por medio de:

- 1) Tabla de Verdad
- 2) Bi-Digraphs
- 3) Diagrama Lógico

### Tabla de verdad.

Una tabla de verdad muestra las combinaciones de los valores de  $X_1, \dots, X_n$  y el valor correspondiente de la expresión. Se utiliza para especificar las salidas de una red lógica en términos de los valores de las variables de entrada.

Para obtenerla, es necesario saber cuantas combinaciones son posibles de realizarse con las variables de entrada de la expresión. Si tenemos una expresión que tiene  $n$  variables y si cada variable puede tomar dos valores, el número total de combinaciones diferentes es:

$$\underbrace{2 \times 2 \times 2 \times 2 \dots \times 2}_{n \text{ variables}} = 2^n$$

La tabla de verdad para  $F(X_1, X_2, \dots, X_n)$  se muestra en la Tabla 1.2.

$X_1 X_2 \dots X_n$	$F(X_1, X_2, \dots, X_n)$
0 0 ... 0	$A_0$
0 0 ... 1	$A_1$
...	...
...	...
1 1 ... 0	$A_{2^{n-2}}$
1 1 ... 1	$A_{2^n-1}$

Tabla 1.2. Tabla de Verdad para  $n$  variables

**Bidigraphs** (Gráfica directa bipartita)

Consiste de dos tipos de nodos interconectados por medio de aristas, de tal forma que de una arista no se conectan dos nodos del mismo tipo. Los nodos utilizados para representar una función de Boole son:

- 1) Nodo circuito que representa una compuerta
- 2) Nodo señal que representa la línea de llegada que interconecta bloques lógicos.

La figura 1.2 muestra el modelo gráfico para la función  $F = (A + BC)(D + (BC)')(D + E)'$ . Los círculos de radio mayor son utilizados para representar una compuerta. Un nombre y la función lógica ejecutada por la compuerta pueden ser asociados al círculo. Los pequeños círculos son utilizados para interconectar las compuertas y el flujo que lleva la señal.

Los bi-digraphs son utilizados para cuando se hace el diseño físico de las redes lógicas.

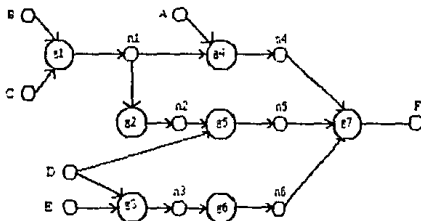


Figura 1.2. Gráfica directa bipartita  
 $F = (A+BC)(D+(BC)')(D+E)'$

**Diagrama lógico.**

El diagrama lógico es una representación gráfica de una o varias funciones de Boole. Utiliza los símbolos mostrados en la tabla 1.1. Un término producto se representa por una compuerta AND, un término suma es representado por una OR, y el complemento de una variable por un inversor.

## I. INTRODUCCION A FUNCIONES DE BOOLE

Las variables de la función son las entradas del circuito, la forma de como interconectar las compuertas es determinada por los paréntesis o por la prioridad de las operaciones a realizar. El orden de las prioridades es:

- 1.- Paréntesis
- 2.- Inversor
- 3.- And
- 4.- Or.

El diagrama lógico de la función  $F = (A + B)C' + D$  está representado en la figura 1.3.

### Clasificación de funciones de Boole.

Una función de Boole es formada por diferentes tipos de expresiones con características que las hacen distintas en su estructura pero que en realidad representan a una misma función.

Sin embargo, aplicando una serie de teoremas del álgebra de Boole se puede obtener una nueva expresión que es equivalente a la función que se encuentra expresada con otros términos.

Las diferentes expresiones de una función son las siguientes:

- a) Suma de productos
- b) Producto de sumas
- c) Mintérminos y/o términos de no importa.
- d) Maxtérminos y/o términos de no importa.

### Suma de productos.

Una expresión es suma de productos cuando los términos producto que la forman son simples variables. Es decir, en cada uno de los términos solamente debe existir el operador AND ( $\cdot$ ). Por ejemplo, la siguiente expresión es una suma de productos.

$$AB' + AEB + ADDE$$

Algunas expresiones pueden tener términos producto de una variable y también son consideradas como suma de productos:

$$A + B + DB + A'B'D$$

Expresiones que no son suma de productos:

$$(A + B) CD + DF + E (A + D)$$

1. INTRODUCCION A FUNCIONES DE BOOLE

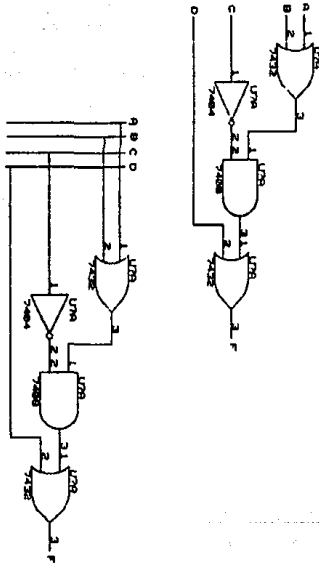


Figura 1.3. Diagrama lógico de  $F = (A + B)C' + D$

**Producto de sumas.**

Una expresión es producto de sumas cuando los términos suma que la forman son simples variables. Es decir, en cada uno de los términos solamente debe de existir el operador OR (+). Por ejemplo, la siguiente expresión es un producto de sumas:

$$(A + B)(C + D' + E)(F + A' + B')$$

Algunas expresiones pueden tener términos suma de una variable y también son consideradas como producto de sumas:

$$(A + C) (C + B + D) (F) \\ (ABC') (A' + B + F')$$

Expresiones que no son producto de sumas:

$$(A + B) (C + D) + EA'$$

**Términos no importa.**

Las X en la Tabla 1.3 indican que la variable de salida F adquiere el valor de 0 ó 1 y no afecta al circuito combinacional. Es decir, éste tipo de términos pueden o no ocurrir.

Cuando se realice la simplificación de una función se debe de tomar en cuenta a los términos de no importa.

Para representar a los términos de no importa en una función de minterminos se anexa  $\Sigma d$ , como lo muestra la siguiente expresión:

$$F = \Sigma(0,2,5) + \Sigma d(3,7)$$

Para representar a los términos de no importa en una función de maxtérminos se anexa  $\pi D$ , como lo muestra la siguiente expresión:

$$F = \pi M(1,5,6) + \pi D(3,7)$$

**Minterminos y/o términos de no importa.**

Dado que una función de Boole es representada por una tabla de verdad, se genera una expresión que está en función de los valores de las variables.

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Tabla 1.3. Tabla de verdad de la función  $A'B'C' + A'BC' + AB'C'$ 

Para expresar la función F en suma de productos se utiliza las combinaciones de las variables para las cuales  $F = 1$ . Por ejemplo, el término  $A'B'C'$  es 1 si  $A = 0$ ,  $B = 0$  y  $C = 0$ . La expresión final es:

$$F = A'B'C' + A'BC' + AB'C' \quad (1.1)$$

Cada uno de estos términos representan a un mintermino. Un mintermino de n variables es un producto de n literales donde cada variable es representada por su complemento si tiene el valor de 0, si el valor es 1 se considera a la variable como verdadera.

El término  $A'B'C'$  es designado como  $m_0$ ,  $A'BC'$  es  $m_1$ , etc. El término  $m_i$  corresponde al renglón de la tabla de verdad (Tabla 1.4).

RENGLON	ABC	MINTERMINO	MAXTERMINO
0	000	$A'B'C' = m_0$	$A + B + C = M_0$
1	001	$A'B'C = m_1$	$A + B + C' = M_1$
2	010	$A'BC' = m_2$	$A + B' + C = M_2$
3	011	$A'BC = m_3$	$A + B' + C' = M_3$
4	100	$A B'C' = m_4$	$A' + B + C = M_4$
5	101	$A B'C = m_5$	$A' + B + C' = M_5$
6	110	$A B C' = m_6$	$A' + B' + C = M_6$
7	111	$A B C = m_7$	$A' + B' + C' = M_7$

Tabla 1.4 Minterminos y Maxterminos para n=3 variables (HIL85)

La expresión 1.1 es representada en términos de  $m_i$ .

$$F(A,B,C) = m_0 + m_2 + m_4 \quad (1.2)$$

En términos generales la expresión 1.1 es abreviada como:

$$F(A,B,C) = \sum (0,2,4) \quad (1.3)$$

Maxtérminos y/o términos no importa.

Para expresar la función F en producto de sumas, se utiliza las combinaciones de las variables para las cuales  $F = 0$ . Por ejemplo, en la tabla 1.3 el término  $A+B+C'$  es 0 si  $A = 0$ ,  $B = 0$  y  $C = 1$ . La expresión final es:

$$F = (A+B+C') (A'+B+C') (A'+B'+C) \quad (1.4)$$

Cada uno de estos términos representan a un maxtérmino. Un maxtérmino de n variables es una suma de n literales donde cada variable es representada por su complemento si tiene el valor de 1, si el valor es 0 se considera a la variable como verdadera.

El término  $A+B+C$  es designado como  $M_0$ ,  $A+B+C'$  es  $M_1$ , etc. El término  $M_i$  corresponde al renglón i de la tabla de verdad (Tabla 1.4).

La expresión 1.4 es representada en términos de  $M_i$ .

$$F(A,B,C) = M_1 + M_2 + M_4$$

En términos generales la expresión 1.4 es abreviada como:

$$F(A,B,C) = \sum (1,2,4)$$

El análisis y diseño de un sistema digital se basa en conceptos básicos como circuito combinacional y circuito secuencial, cuyo fundamento se encuentra en la lógica de Boole.

### Circuito combinacional.

Un circuito combinacional está formado por variables de entrada, compuertas lógicas y variables de salida (MORBS). Tiene como función realizar operaciones lógicas. La característica principal del circuito es que las salidas dependen de la combinación que exista entre las señales de entrada en un instante sin importar sus valores anteriores, y de la combinación que exista entre las compuertas.

Este tipo de circuitos puede ser descrito por m funciones de Boole, una para cada variable de salida:

$$Z_m = F_m (X_1, X_2, \dots, X_n)$$

## 1. INTRODUCCION A FUNCIONES DE BOOLE

Cada una de las funciones de salida es expresada en términos de las  $n$  variables de entrada.

El diagrama a bloques del circuito combinacional se muestra en la figura 1.4.



Figura 1.4. Diagrama a bloques de un circuito combinacional

En el desarrollo de circuitos combinacionales existen dos pasos que son importantes para los fines de este estudio.

- Obtener de una función de boole simplificada para cada salida.
- Realizar del diagrama lógico.

### Circuito secuencial.

Un circuito secuencial está formado por un circuito combinacional y elementos que almacenan información binaria en un tiempo determinado. Un diagrama de bloques de este tipo de circuitos se muestra en la figura 1.5 (MOR85).

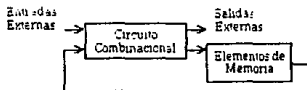


Figura 1.5. Diagrama a bloques de un circuito secuencial

Como se puede observar en el diagrama a bloques, el sistema es realimentado por los elementos de memoria, que tienen almacenado el valor de las combinaciones anteriores. Las nuevas señales de salida y el nuevo estado de los elementos de memoria se generan con el valor almacenado y con las señales de entrada.



Dentro de los circuitos secuenciales existe la siguiente clasificación:

**Circuitos secuenciales sincrónicos**  
**Circuitos secuenciales asincrónicos**

Un circuito secuencial sincrónico es un sistema cuyo comportamiento puede definirse a partir del conocimiento de sus señales en instantes discretos de tiempo. Como elementos de memoria se emplean flip-flops que cambian un bit de información cuando es afectado por un pulso de reloj.

Un circuito secuencial asincrónico es un sistema en el cual no se puede determinar su comportamiento, porque depende del orden en que cambien las señales de entrada en cualquier instante de tiempo. Como elementos de memoria se emplean compuertas con rutas de realimentación que lo hacen ser inestable, limitando su uso en el diseño de sistemas.

**PLANTEAMIENTO DEL PROBLEMA****II.1. DEFINICION DEL PROBLEMA.**

Para establecer si el diseño de un sistema digital es eficiente, el criterio que se utiliza es verificar que sea el más simple y que realice el objetivo deseado.

El problema es el desarrollar un programa para microcomputadoras que permita obtener las expresiones lógicas con menos términos, así como la graficación e impresión del diagrama lógico.

Para tal efecto se necesita de una disciplina que permita identificar las causas y problemas que existen para el desarrollo del sistema. Esta disciplina es conocida como Ingeniería de Programación. El modelo a utilizar es el Ciclo de vida clásico.

El minimizar o simplificar una función de Boole requiere de cierto proceso que se aplica directamente a la expresión de Boole obteniéndose un nuevo modelo matemático cuyo comportamiento cubre todas las necesidades del sistema.

Este tipo de proceso es aplicado por medio de varios métodos que son analizados, para poder elegir el más adecuado, de acuerdo a los siguientes criterios:

- a) Existe algún algoritmo para aplicar el método.
- b) Seguridad en la nueva expresión.

Los métodos que a estudiar son:

- 1) Algebra de Boole
- 2) Mapa de Karnaugh
- 3) Quine McCluskey

## II.2. ALTERNATIVAS DE SOLUCION.

## II.2.1. ALGEBRA DE BOOLE.

El cimiento de la simplificación está en conocer las propiedades fundamentales del sistema de Boole. El álgebra de Boole es un sistema matemático que está formado por un conjunto B de dos elementos distintos y dos operadores binarios. Los elementos pueden ser representados arbitrariamente por cualquier símbolo; en el área digital se utiliza el 0 y el 1.

$$B = \{0,1\}$$

Cada uno de estos elementos es referenciado como un valor a una constante. El valor de un elemento es tomado a partir de una variable, una constante tiene la característica de que nunca va a cambiar su valor, ante cualquier situación.

Para poder representar a una variable o una constante se utiliza un alfabeto que es un conjunto finito de símbolos tales como letras, dígitos y símbolos especiales.

$$\text{Alfabeto} = \{A,B,C,\dots,Z, a,b,c,\dots,z,0,1,\dots,9,-,/,\# \text{ etc.}\}$$

En algunos casos una variable x puede tener asignado el valor  $\alpha$  en un instante de tiempo y en otro el valor de  $\beta$  o simplemente simbolizar algún elemento del conjunto B. Cuando dos variables poseen el mismo valor para cualquier instante, se dice que ambas variables son iguales y se puede representar cualquier ocurrencia de la variable A con el símbolo B y viceversa. Matemáticamente esto es conocido como la ley de sustitución  $A = B$ .

Como todo conjunto, cada elemento del conjunto B está relacionado a otro único miembro de B llamado complemento y es representado por una literal y un apóstrofe. Una literal es un símbolo que representa una variable o un complemento. Una o más literales en unión con un operador binario representan términos producto y términos suma.

Las operaciones básicas del álgebra de Boole son la AND ( $\wedge$ ), OR (+) y el complemento ('). Los símbolos y tabla de verdad de cada operación se encuentran en la tabla 1.1.

## Teoremas básicos.

Operaciones con 0 y 1:

$$\begin{array}{ll} X + 0 = X & (2.1) \\ X + 1 = 1 & (2.2) \end{array} \quad \begin{array}{ll} X \wedge 1 = 1 & (2.1D) \\ X \wedge 0 = 0 & (2.2D) \end{array}$$

Ley Idempotent:

$$X + X = X \quad (2.3) \quad X \cdot X = X \quad (2.3D)$$

Ley Involución:

$$(X')' = X \quad (2.4)$$

Ley del complemento:

$$X + X' = 1 \quad (2.5) \quad X \cdot X' = 0 \quad (2.5D)$$

Cada uno de estos teoremas son fáciles de demostrar, asignádole a  $X$  valores de 0 y 1.

Por ejemplo demostrar el teorema  $X + 0 = X$

si  $X = 0$ ,  $0 + 0 = 0$  y si  $X = 1$ ,  $1 + 0 = 1$

Para el desarrollo del álgebra de Boole se utiliza diferentes conjuntos de axiomas que deben de cumplir con las siguientes características:

- 1) Los axiomas de un conjunto deben de ser consistentes, es decir, que no existan contradicciones en el desarrollo lógico del álgebra por deducción.
- 2) Deben de ser mutuamente independientes y libres de redundancia.
- 3) Deben de ser simples para facilitar el desarrollo del álgebra.

### Axiomas del álgebra de Boole.

Los operadores son conmutativos

Para cada  $X \in B$ ,  $Y \in B$

$$XY = YX \quad (2.6) \quad X + Y = Y + X \quad (2.6D)$$

Los operadores son asociativos

Para cada  $X, Y, Z \in B$

$$\begin{aligned} (X + Y) + Z &= X + (Y + Z) \\ &= X + Y + Z \end{aligned} \quad (2.7)$$

$$(XY)Z = X(YZ) = XYZ \quad (2.7D)$$

Los operadores son distributivos sobre cada uno.

$$X(Y + Z) = XY + XZ \quad (2.8)$$

$$X + YZ = (X + Y)(X + Z) \quad (2.8D)$$

**Leyes de Morgan.**

$$(X + Y)' = X'Y' \quad (2.9) \quad (XY)' = X' + Y' \quad (2.9D)$$

Las Leyes de Morgan se generalizan a  $n$  variables

$$(X1+X2+X3+...+Xn)' = (X1' \cdot X2' \cdot X3' \dots Xn')$$

$$(X1 \cdot X2 \cdot X3)' = X1' + X2' + X3' + \dots + Xn'$$

Combinando ambas formas de la Ley de Morgan se crea una regla que se aplique en un solo paso.

Para complementar una expresión de Boole es necesario reemplazar cada variable por su complemento, los 0 por 1, los 1 por 0, el operador + por el operador  $\cdot$  y el operador  $\cdot$  por el operador +.

Simbólicamente esta regla se representa:

$$[F(X1, X2, \dots, Xn, 0, 1, \cdot, +)]' = F(X1', X2', \dots, Xn', 1, 0, +, \cdot) \quad (2.10)$$

**Dualidad.**

Dada una expresión de Boole, su dual se forma reemplazando los 0 por 1, los 1 por 0, el operador  $\cdot$  por el operador + y el operador + por el operador  $\cdot$ . Simbólicamente la regla para formar el dual de una expresión es:

$$[F(X1, X2, \dots, Xn, 0, 1, \cdot, +)]' = F(X1, X2, \dots, Xn, 1, 0, +, \cdot) \quad (2.11)$$

Cuando se aplican las Leyes de Morgan o se requiere del dual de una expresión hay que preservar la jerarquía que existe entre las operaciones adicionando o eliminando paréntesis.

**Teoremas de simplificación.**

Los siguientes teoremas son útiles para la minimización de expresiones de Boole.

$$xy + xy' = x \quad (2.12) \quad (x + y)(x + y') = x \quad (2.12D)$$

$$x + xy = x \quad (2.13) \quad x(x + y) = x \quad (2.13D)$$

$$(x + y')y = xy \quad (2.14)$$

$$xy' + y = x + y \quad (2.14D)$$

### Teorema del consenso.

Dado un par de términos en el cual una variable aparece en un término y en el otro su complemento, el término del consenso se forma multiplicando las variables restantes de cada término. Por ejemplo, el consenso de  $AB$  y  $A'D$  es  $BD$ , el consenso de  $AB'D$  y  $A'B'D'$  es cero.

Este teorema es definido como:

$$XY + Z' + YZ = XY + X'Z \quad (2.15)$$

$$(X+Y)(X'+Z)(Y+Z) = (X+Y)(X'+Z) \quad (2.15D)$$

Para el dual de este teorema se adicionan las variables restantes de cada término.

El teorema del consenso se aplica para eliminar términos redundantes.

Por ejemplo, los términos consenso de la función  $F = A'B' + AC + BC' + B'C + AB$ , son:

El consenso de  $A'B'$  y  $AC$  es  $B'C$  y de  $AC$  y  $BC'$  es  $AB$ . Por lo tanto:

$$F = A'B' + AC + BC'$$

Algunas veces el teorema del consenso no se puede aplicar directamente para minimizar una expresión. En estos casos es conveniente adicionar términos utilizando el teorema del consenso y utilizarlos para eliminar otros términos.

$$\text{Por ejemplo } F = \underbrace{ABCD + B'CDE}_{ACDE} + \underbrace{A'B' + BCE'}_{A'CE'}$$

Los términos que se pueden adicionar son  $ACDE$  y  $A'CE'$ . Si adicionamos  $ACDE$ , se elimina  $ABCD$ , combinando  $ACDE$  y  $BCE'$ , y se elimina  $B'CDE$  combinando  $A'B'$  y  $ACDE$ .

$$F = ABCD + B'CDE + A'B' + BCE' + ACDE$$

$$\text{Finalmente } F = A'B' + BCE' + ACDE$$

El término  $A'CE'$  no elimina algún término.

**Simplificación algebraica.**

La simplificación de funciones de Boole se puede realizar de cuatro formas, combinando términos, eliminando términos, eliminando literales y adicionando términos redundantes.

**Combinación de términos.** Se usa el teorema  $XY + XY' = X$  para combinar dos términos.

Ejemplo:  $ABC'D' + ABCD' = ABD'$  si  $X = ABD'$ ,  $Y = C$ .

Los dos términos a combinar deben contener las mismas variables y una de las variables debe estar completamente en un término y en el otro no. Si  $X + X = X$ , un término puede ser duplicado y combinado con dos o mas términos.

Ejemplo:  $AB'C + ABC + A'BC = AB'C + ABC + ABC + A'BC$   
 $= AC(B+B') + BC(A+A')$   
 $= AC + BC$

**Eliminando términos.** Se usa el teorema  $X + XY = X$  para eliminar términos redundantes si es posible; se aplica el teorema de consenso  $XY + X'Z + YZ = XY + X'Z$

Ejemplo:  $A'B + A'BC = A'B$  si  $X = A'B$ ,  $Y = C$

$A'BC' + BCD + A'BD = A'BC' + BCD$  si  $X = C$ ,  $Y = BD$ ,  $Y Z = A'B$

**Eliminando literales.** Se usa el teorema  $X + X'Y = X + Y$  para eliminar literales redundantes antes de aplicar el teorema hay que factorizar.

Ejemplo:  $A'B + A'B'C'D' + ABCD' = A'(B+B'C'D') + ABCD'$   
 $= A'(B + C'D') + ABCD'$   
 $= A'B + A'C'D' + ABCD'$   
 $= B(A' + ACD')$   
 $= B(A' + CD')$   
 $= BA' + BCD' + A'C'D'$

**Adicionando términos redundantes.** Los términos redundantes son introducidos de distintas formas, adicionando  $XX'$ , multiplicando por  $(X + X')$ , adicionando  $YZ$  a  $XY + X'Z$ , o adicionando  $XY$  a  $X$ . La razón por la cual son adicionados a la expresión es porque se combinan con otros términos para eliminar nuevos términos.

Ejemplo:  $WX + XY + X'Z' + WY'Z'$

Aplicando el teorema del consenso obtenemos:

$WX + XY + X'Z' + WY'Z' + WZ'$  (eliminando  $WY'Z'$ )

$$WX + XY + X'Z' + WZ' \quad (Y'+1)$$

$$WX + XY + X'Z' + WZ'$$

En el siguiente ejemplo se aplicarán las cuatro reglas para minimizar una función.

$$F = A'B'D + ABC' + AB'CD + BC + A'CD + BCD$$

Factorizando:

$$F = B'D(A'+AC) + BC(1+D) + ABC' + A'CD$$

$$\text{Si } X=A', Y=C, \text{ y aplicando } X+X'Y = X+Y$$

$$F = B'D(A'+C) + BC + ABC' + A'CD$$

$$F = B'DA' + B'DC + BC + ABC' + A'CD$$

Aplicando el teorema del consenso  $XY + X'Z + YZ = XY + X'Z$  y si  $X=B, Y=C, Z=A'D$

$$F = B'DA' + B'DC + BC + ABC'$$

Si adicionamos el término consenso  $CD$  y factorizamos

$$F = B'DA' + CD(B'+1) + BC + ABC'$$

$$F = B'DA' + CD + BC + ABC'$$

Si factorizamos y  $X=C, Y=A$

$$F = B'DA + CD + B(AC'+C)$$

$$F = B'DA' + CD + B(A+C)$$

Por lo tanto

$$F = A'B'D + AB + BC + CD$$

## II.2.2. MAPA DE KARNAUGH.

El mapa de Karnaugh es un método gráfico que permite minimizar funciones de Boole para encontrar la mínima expresión en suma de productos o en producto de sumas.

Una mínima expresión en suma de productos debe de tener:

- a) Un mínimo número de términos.



- b) Todas las expresiones que tienen el mismo número de términos, deben de tener un mínimo número de literales.

Una mínima expresión en producto de sumas debe de tener:

- a) Un mínimo número de factores.  
 b) Todas las expresiones que tienen el mismo número de factores, deben de tener un mínimo número de literales.

Debido a que el método es aplicable a cualquier tipo de funciones, se realiza la simplificación de funciones en suma de productos dado que para los otros tipos de funciones el proceso es idéntico, salvo pequeños cambios que se detallarán en el momento adecuado.

El mapa de Karnaugh es una transformación de un diagrama de Venn. Considere el diagrama de la figura 2.1a. Las variables A y B representan subdivisiones de un conjunto universal. En la figura 2.1b se observa que cada una de las subdivisiones representan la intersección de AB, A'B, AB' y A'B'. Cada una de éstas intersecciones son los minterminos de una función de Boole de dos variables. Cada subdivisión es renombrada como minterminos m0, m1, m2, y m3 (figura 2.1c). Es necesario ajustar cada una de las áreas para que todos sean iguales. Ahora la mitad del diagrama representa a la variable A y la otra a B. (figura 2.1d). Las figuras 2.1e y 2.1f son dos formas de representar a un mapa de Karnaugh.

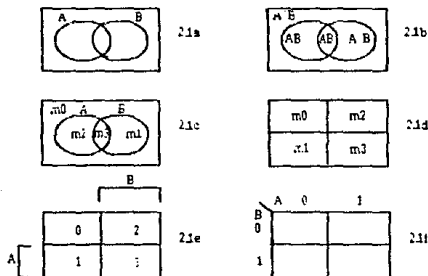


Figura 2.1. Diagrama de Venn y mapa para dos variables

Los 0 y 1 que se encuentran en las columnas y renglones de la figura 2.1f representan los valores de las variables A y B. Esta secuencia de valores es determinada por el Código Gray o Código Reflejado, el cual varía en un sólo bit.

Código Gray	Código Decimal
A B	A
0 0	0
0 1	1
1 1	2
1 0	3

Código Gray	Código Decimal
A B C	A
0 0 0	0
0 0 1	1
0 1 1	2
0 1 0	3
1 1 0	4
1 1 1	5
1 0 1	6
1 0 0	7

Tabla 2.1. Código Gray para 2 y 3 variables

En la tabla 2.1 se muestra el Código Gray para 2 y 3 variables. Los renglones de una tabla de verdad representan a un mintermino del mapa que es identificado por un pequeño cuadro. El número de cuadros depende del número de variables de la función a simplificar.

En la figura 2.2 se muestra los mapas de Karnaugh para 2, 3, 4, 5, y 6 variables.

El minimizar un mapa con más de 6 variables resulta muy impráctico y crea muchos problemas.

**Representación de una función en un mapa.**

Las representaciones más usuales son:

- a) Tabla de verdad
- b) Funciones incompletas

- c) Minterminos  
 d) Maxtérminos  
 e) Minterminos o Maxtérminos con términos de no importa.

A	0	1
B	1	2
1	1	3

Mapa de dos variables

A	B	00	01	11	10
0	1	2	3	4	5
1	6	7	8	9	10

Mapa de tres variables

A	B	00	01	11	10
0	0	4	12	8	
0	1	5	13	9	
1	0	6	14	10	
1	1	7	15	11	

Mapa de cuatro variables

A	B	00	01	11	10
0	0	4	12	8	
0	1	5	13	9	
1	0	6	14	10	
1	1	7	15	11	

Mapa de cinco variables

A	B	000	001	011	010
0	0	4	12	8	
0	1	5	13	9	
1	0	6	14	10	
1	1	7	15	11	

A	B	100	101	111	110
0	0	16	20	24	28
0	1	17	21	25	29
1	0	18	22	26	30
1	1	19	23	27	31

A	B	000	001	011	010
0	0	16	20	24	28
0	1	17	21	25	29
1	0	18	22	26	30
1	1	19	23	27	31

A	B	100	101	111	110
0	0	24	28	32	36
0	1	25	29	33	37
1	0	26	30	34	38
1	1	27	31	35	39

A	B	000	001	011	010
0	0	32	36	40	44
0	1	33	37	41	45
1	0	34	38	42	46
1	1	35	39	43	47

A	B	100	101	111	110
0	0	48	52	56	60
0	1	49	53	57	61
1	0	50	54	58	62
1	1	51	55	59	63

Mapa de seis variables

Figura 2.2. Mapas de Karnaugh para 2,3,4,5 Y 6 variables (IRWBO)

## Tabla de verdad.

Una tabla de verdad representa a una función de Boole y si cada renglón de la tabla representa a un mintermino, el procedimiento a seguir es trasladar cada uno de los minterminos que tienen una salida  $F = 1$  de la tabla de verdad al mapa. En la figura 2.3 se muestra la tabla de verdad de una función y su mapa.

A	B	F	Mintermino
0	0	1	$m_0$
0	1	1	$m_1$
1	0	0	$m_2$
1	1	1	$m_3$

A	B	0	1
0	0	1	1
1	0	0	0
1	1	1	1

Figura 2.3. Tabla de verdad y su mapa correspondiente

## Funciones incompletas.

Se considera a una función incompleta cuando los términos suma o producto de los cuales está formada no incluyen a todas las variables que intervienen en la función. Por ejemplo:

$$F(A,B,C) = A'B + B' + A'BC$$

La forma de representar esta función es la siguiente:

- 1) El término  $A'B$  es 1 cuando  $A'B = 01$ , así que colocamos un 1 en las celdas en donde  $A'B = 01$ . Figura 2.4a.
- 2) El término  $B'$  es 1 cuando  $B' = 0$ , así que colocamos un 1 en las celdas en donde  $B' = 0$ . Figura 2.4b.
- 3) El término  $A'BC$  es 1 cuando  $A'BC = 011$ . Si ya existe un 1 en la celda correspondiente, no es necesario colocar un segundo 1 porque  $X + X = X$ . Figura 2.4c.

A	B	C	01	11	10
0	0	1	1	0	0
1	0	1	1	0	0

2.4a

A	B	C	01	11	10
0	0	1	1	0	0
1	0	1	1	0	0
1	0	0	1	0	0
1	1	0	1	0	0

2.4b

A	B	C	01	11	10
0	0	1	1	0	0
1	0	1	1	0	0
1	0	0	1	0	0
1	1	0	1	0	0
1	1	1	1	1	0

2.4c

A	B	C	01	11	10
0	0	1	1	0	0
1	0	1	1	0	0
1	0	0	1	0	0
1	1	0	1	0	0
1	1	1	1	1	0

2.4d

Figura 2.4. Mapa de Karnaugh de  $F(A,B,C) = A'B+B'+A'BC$

**Mintérminos.**

Si la función de Boole se encuentra expresada en suma de productos, el procedimiento a seguir es colocar un 1 en la celda correspondiente. Por ejemplo:

$$F(A,B,C) = \Sigma m(0,2,4,7)$$

La representación de los mintérminos  $m_0$ ,  $m_2$ ,  $m_4$  y  $m_7$  en el mapa, se observa en la figura 2.5.

Cada una de las celdas que se encuentran vacías representan a un mintérmino que no se encuentra en la función y usualmente toman el valor de 0.

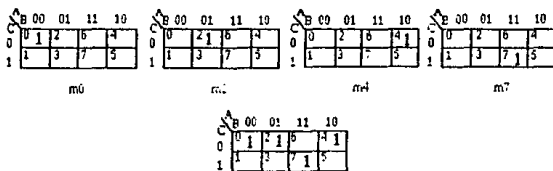


Figura 2.5. Mapa de Karnaugh de  $F(A,B,C) = \Sigma m(0,2,4,7)$

**Maxtérminos.**

Si la función de Boole se encuentra expresada en un producto de sumas, el procedimiento a seguir es colocar un 0 en la celda correspondiente. Por ejemplo:

$$F(A,B,C) = \Pi M(0,1,4,6)$$

La representación de los maxtérminos  $M_0$ ,  $M_2$ ,  $M_4$  y  $M_7$  en el mapa, se observa en la figura 2.6.

Cada una de las celdas que se encuentran vacías representan a un maxtérmino que no se encuentra en la función y usualmente toman el valor de 1.

## 11.2. ALTERNATIVAS DE SOLUCION

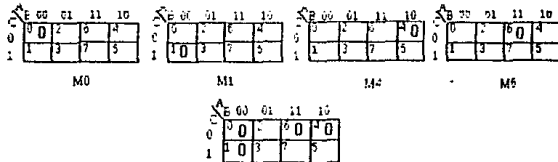


Figura 2.6. Mapa de Karnaugh de  $F(A,B,C) = \pi M(0,1,4,6)$

Mintérminos o maxtérminos con términos de no importa.

Un término de no importa puede ser incluido en el diseño si ayuda en la simplificación del circuito lógico. Los términos no importa pueden ocurrir cuando ciertas combinaciones de entrada nunca son aplicadas a la función de Boole o en el caso en que todas las combinaciones de entrada ocurren pero la salida es requerida como 1 ó 0 para ciertas combinaciones.

El procedimiento a seguir es colocar una X en la celda correspondiente. Como se ilustra en la figura 2.7.



$$F = \text{Em}(0,1,3) + \text{Ed}(2,4)$$

$$F = \pi M(5,6,7) \dagger \pi D(2,4)$$

Figura 2.7. Mapa con términos no importa

Los términos no importa son tratados en la simplificación como 1's en sumatoria de mintérminos y como 0's en producto de maxtérminos.

### Propiedades del mapa de Karnaugh.

Cada una de las siguientes propiedades dependen de la posición que ocupen cada una de las celdas que contengan un 1, para poder determinar si son términos adyacentes (LEEB2).

Un término es adyacente a otro si difieren en el valor de una variable que los forma.

Las combinaciones de unos se deben de agrupar en potencias de dos. En la figura 2.8 se observa el número de unos que se pueden combinar y las variables que se eliminan. Un punto muy importante es el obtener el menor número de grupos que cubra a todos los términos de la función.

Grupo de celdas	Variables eliminadas
2	1
4	2
8	3
16	4

Figura 2.8. Combinación de 1's para eliminar variables

### Propiedad 1.

Cualquier par de celdas adyacentes que contengan un 1 pueden ser combinadas y formar un término con una variable eliminada. Se puede utilizar un término varias veces para simplificar la función. En la figura 2.9 se muestra un ejemplo.

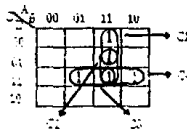


Figura 2.9. Agrupación de dos 1's en distintas formas

## Propiedad 2.

Cuatro celdas que contengan un 1 pueden ser agrupadas para formar un término y así dos variables son eliminadas. En la figura 2.10 se muestran 6 formas distintas para agrupar unos.

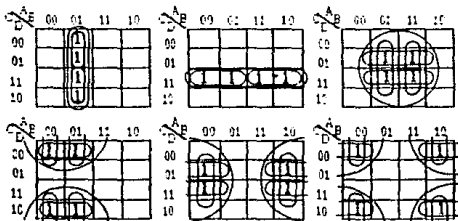


Figura 2.10. Agrupación de cuatro 1's en distintas formas

## Propiedad 3.

Si ocho celdas contienen un 1 y son adyacentes se combinan en un sólo término y tres variables son eliminadas. En la figura 2.11 se muestran cuatro formas distintas para agrupar unos.

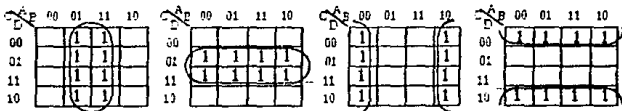


Figura 2.11. Agrupación de ocho 1's en distintas formas

La forma de encontrar la mínima expresión es emplear la propiedad adecuada para agrupar celdas. Y para lograrlo se puede utilizar un procedimiento.



Antes de mencionar este procedimiento se definen algunos conceptos que serán útiles para cuando se aplique el método de simplificación.

### Implicante.

Un implicante es cada término y grupos de 1's que puedan ser combinados para formar un término producto. En la figura 2.12 se observa una serie de implicantes de una función.

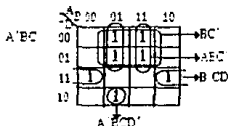


Figura 2.12. Implicantes de una función

### Primo implicante.

Un implicante es llamado primo implicante si no puede ser combinado con otro término para eliminar una variable o si no es un subconjunto de otro implicante de la función.

De la figura 2.12 el término  $BC'$ ,  $B'CD$  son primos implicantes porque no se pueden combinar con otros términos. En cambio  $A'BC'D'$  no es primo implicante porque combinándose con  $A'BCD'$  elimina a la variable  $C$ .

### Primo implicante esencial.

Si un mintermino es cubierto por sólo un primo implicante, se dice que el primo implicante es esencial y puede ser incluido en la mínima expresión.

Se puede observar en la figura 2.13 cada uno de los primos implicantes del mapa. El término  $BD$  es un primo implicante esencial porque el mintermino  $m15$  se encuentra incluido exclusivamente por el término  $BD$ . Por otro lado el término  $A'B$  no es esencial porque cada uno de sus minterminos pueden estar incluidos en cualquier otro implicante. Los primos implicantes  $A'C$ ,  $A'C'D'$  son esenciales.

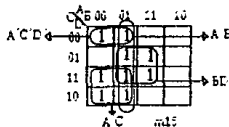


Figura 2.13. Primos implicantes esenciales

Procedimiento para obtener una mínima expresión de un mapa.

- 1) Elegir un mintérmino que no ha sido cubierto.
- 2) Encontrar todos los 1 y X adyacentes a ese mintérmino.
- 3) Si un solo término cubre el mintérmino y todos los 1's y X's adyacentes, entonces el término es un primo implicante esencial y se incluye en la mínima expresión.
- 4) Repetir los pasos 1, 2 y 3 hasta que todos los primos implicantes esenciales hayan sido elegidos.
- 5) Encontrar un mínimo conjunto de primos implicantes los cuales cubran el resto de 1's sobre el mapa. Si existe más de uno hay que elegir el que posea el mínimo número de literales.

Ejemplo 1. Determine la mínima expresión en suma de productos de F.

Como se observa en la figura 2.14 m0 es adyacente a m8, m2, y m10, siendo cubiertos solamente por el término B'D'; así que B'D' es un primo implicante esencial. El mintérmino m5 es adyacente a m7 y como es cubierto sólo por el término A'BD, A'BD es otro primo implicante esencial; m3 es adyacente a m7 y m2, al no ser cubiertos por el mismo término, no es un primo implicante esencial. El resto de unos son cubiertos por los primos implicantes ABC, A'B'C ó A'CD.

Por lo tanto la mínima expresión es :

$$F = B'D' + A'BD + ABC + A'B'C$$

## II.2. ALTERNATIVAS DE SOLUCION

$$F = B'D' + A'BD + ABC + A'CD$$

En este ejemplo existen dos mínimas expresiones y como ambas poseen el mismo número de literales, no importa cual se elija.

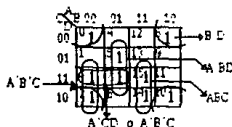


Figura 2.14. Mapa para la función  
 $F = B'C'D' + BCD + ACD' + A'B'C + A'BC'D$

Ejemplo 2. Encontrar la mínima expresión en suma de productos de F.

$$F(A,B,C,D) = E_m(2,3,4,7,9,11,12,13,14) + E_d(1,10,15)$$

En la figura 2.15 m4 es adyacente a m12 y son cubiertos por el término  $BC'D'$ , así que  $BC'D'$  es un primo implicante esencial. El mintermino m13 es adyacente a m9 y X15, siendo cubiertos por el término AD, por lo tanto AD es un primo implicante esencial; m7 es adyacente a m3, X15 y son cubiertos por el término CD, CD es un primo implicante esencial; m2 es adyacente a m3, X10 y el término que los cubre es  $B'C$  convirtiéndose en primo implicante esencial. Finalmente el resto de 1's es cubierto por AB.

La mínima expresión es :

$$F = AD + B'C + BC'D' + CD + AB$$

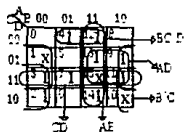


Figura 2.15. Mapa para la función  
 $F = E_m(2,3,4,7,9,11,12,13,14) + E_d(1,10,15)$

Mapa de Karnaugh para cinco y seis variables.

Un mapa de cinco variables es representado por dos mapas de cuatro variables como se muestra en la figura 2.16.

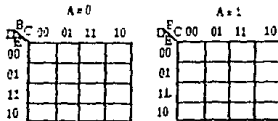


Figura 2.16. Mapa de Karnaugh para 5 variables

Uno de los mapas representa a una variable con el valor de 0 y el otro mapa cuando vale 1.

Para una función de seis variables un mapa de Karnaugh es representado por cuatro mapas de cuatro variables, cada uno es la combinación de dos variables. En la figura 2.17 se observa esta representación.

Las propiedades de los mapas de Karnaugh de cuatro variables se aplican en las funciones de cinco y seis variables, con la condición de que las variables del mapa ocupen la misma posición relativa.

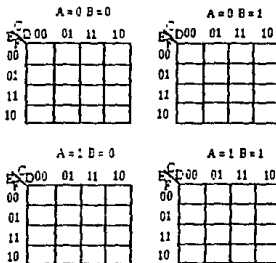


Figura 2.17. Mapa de Karnaugh para seis variables

Ejemplo 3. Encontrar la mínima expresión de la función:

$$F(A,B,C,D,E) = \Sigma m(4,5,6,7,13,15,20,21,22,23,25,27,29,31)$$

En la figura 2.18 el mintermino m4 es adyacente a m5 y a m20 y dado que son cubiertos por el término  $B'C$ ,  $B'C$  es un primo implicante esencial; m13 es adyacente a m5, m15 y a m29 y el término que los cubre es CE, por lo tanto CE es un primo implicante esencial. Finalmente el término ABE es otro primo implicante esencial porque cubre a m29, m3, m31 y m27. La expresión mínima es :  $F = ABE + B'C + CE$

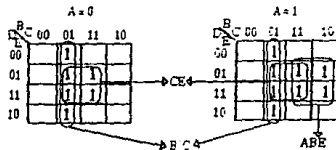


Figura 2.18. Mapa de Karnaugh para el ejemplo 3.

Ejemplo 4. Encontrar la mínima expresión de la función:

$$F(A,B,C,D,E) = \Sigma m(0,1,2,3,7,13,15,16,17,18,19,29,31,35,39,47,45,51,55,59,63) + \Sigma d(5,21,27,33,37,48,61,54)$$

Obteniendo cada uno de los primos implicantes esenciales, tenemos que la función minimizada es:  $F = A'C'D' + CDF + B'C'F + ABEF$

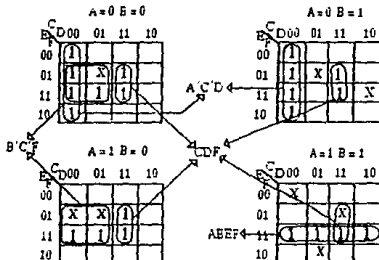


Figura 2.19. Mapa de Karnaugh del ejemplo 4.

## Mapa de Karnaugh en producto de sumas

El algoritmo a seguir es :

- 1) Llenar el mapa con sus correspondientes valores.
- 2) Agrupar los 0's y X's aplicando la propiedad adecuada.
- 3) Escribir la función en producto de sumas, complementando cada una de las variables.

Ejemplo 5. Utilizando el mapa de la figura 2.20 que se muestra a continuación, obtener la mínima expresión en producto de sumas.

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

Arrows pointing to the 0s in the map:  
 - Top-right 0 (M9) is labeled  $A' + B$   
 - Middle-right 0 (M5) is labeled  $MB + C + D'$   
 - Bottom-right 0 (M13) is labeled  $MA' + D$

Figura 2.20. Mapa de Karnaugh del ejemplo 5

El término suma  $A' + B$  se genera porque el maxtérmino M9 es adyacente a M8 y M11 siendo todos cubiertos, si la variable A es 1 se debe tomar como  $A'$  y como B es 0 se considera como B.

La mínima expresión es :

$$F = (A' + B)(A' + D)(B + C + D')$$

## II.2.3. QUINE McCLUSKEY.

Es un método sistemático con pasos bien definidos que realiza la búsqueda de términos que son seleccionados para encontrar una función simplificada.

Cabe recordar que un primo implicante es un término producto que puede ser combinado con otros para producir un término con menos literales.

Y un primo implicante esencial es un primo implicante que cubre por lo menos un mintermino de la función que no es cubierto por otro primo implicante.

El obtener una función mínima por medio de este método está basado en la aplicación del criterio de costo de la función, que reduce el número de compuertas. A cada uno de los primos implicantes de la función se le asigna un costo que depende del número de variables que tenga el implicante. Si un primo implicante tiene cuatro variables, su costo es demasiado alto comparado con otro que sólo posee tres, dos o una variable. Por lo tanto, se considera a una función mínima cuando se tenga el menor costo en su realización física.

El método a seguir para obtener una mínima expresión se enumera a continuación :

1) Listar todos los minterminos de la función a ser minimizada en su representación binaria separándolos en grupos de acuerdo al número de 1's.

2) Realizar una búsqueda exhaustiva para los minterminos adyacentes combinándolos para formar una nueva lista de primos implicantes.

3) Construir una tabla que va a tener horizontalmente cada uno de los minterminos y verticalmente todos los primos implicantes de la función. Colocar una X en cada uno de los minterminos que son cubiertos por cada uno de los primos implicantes.

4) Seleccionar el mínimo número de primos implicantes que cubran todos los minterminos de la función de Boole y que posean el menor costo.

Por medio del siguiente ejemplo se explica detalladamente cada uno de estos pasos.

Ejemplo 1. Encontrar la mínima expresión de la función:

$$F(A,B,C,D) = \Sigma m(0,2,3,5,7,8,10,14,15)$$

Como indica el paso 1 los minterminos deben ser agrupados de acuerdo al número de unos que posean. Se genera la tabla 2.2.

Una vez creada esta tabla se inicia el paso 2. Para poder combinar dos términos es necesario que difieran en una sola variable. Bajo esta condición solo se pueden combinar los grupos que difieran en un 1. Por ejemplo, el grupo 0 solo se puede combinar con el grupo 1 y el grupo 1 con el grupo 2 y así sucesivamente hasta terminar con todos los grupos. Por cada variable que sea eliminada se coloca en su posición un "-". Cuando se han realizado todas las combinaciones se genera el listado 2 de la tabla 2.3.

Mintérmino	ABCD	No. de unos	Mintérmino	ABCD	
0	0000	0	0	0000	Grupo 0
2	0010	1	2	0010	Grupo 1
3	0011	2	3	1000	
4	0101	3	4	0011	Grupo 2
5	0111		5	0101	
7	1000	6	1010		
8	1010	7	10	0111	Grupo 3
10	1011	8	14	1110	
14	1111	9	15	1111	

Tabla 2.2. Agrupación de mintérminos de acuerdo a su número de unos

La línea que se encuentra entre cada una de estas combinaciones sirve para separar cada uno de los grupos. Cada mintérmino de los listados que es combinado, debe de tener una marca (✓) que indique que ya fue seleccionado para formar parte del siguiente listado. Un mintérmino puede ser combinado más de una vez sin importar que se encuentre ya marcado.

Listado 1		Listado 2		Listado 3		
Mintérminos	ABCD	Mintérminos	ABCD	Mintérminos	ABCD	
0	0000	0,2	00_0	0,2,8,10	_00_	P1
2	0010	3,3	_000			
3	1000	2,3	001_			P2
4	0011	2,10	_010			P3
5	0101	8,10	10_0			P4
10	1010	3,7	0_11			P5
7	0111	5,7	01_1			P6
14	1110	10,14	1_10			P7
15	1111	7,15	_111			
		14,15	11_1			

Tabla 2.3. Listados de primos implicantes

En el listado 2 el primer elemento indica que el mintérmino 0, y 2 fueron combinados eliminando a la variable C, algebraicamente sería lo siguiente:

$$\begin{aligned}
 A'B'C'D' + A'B'CD' &= A'B'D'(C'+C) \\
 &= A'B'D' \text{ ó } A'B'_D' \text{ ó } 00_0
 \end{aligned}$$

Todos los elementos del listado 2 son generados de la misma manera y una vez creado hay que generar el listado 3. En este momento



persiste la condición de que dos términos se pueden combinar si difieren en una variable y deben de coincidir en la posición del símbolo "\_", para el listado 3.

Los únicos mintérminos que se pueden combinar son el 0,2 y 8,10 generando en el listado 3 a 0,2,8,10. Una vez creado el listado 3 todos los términos que no fueron marcados son primos implicantes y etiquetados desde P1...P7.

Cada uno de estos primos implicantes forman a la función de Boole, pero como se necesita tener la mínima expresión se crea una tabla de primos implicantes para seleccionar aquellos que cubren a la función, iniciandose el paso 3.

Etiqueta	Término Producto	Mintérminos						Costo
		0	2	3	7	10	14	
* P1	0,2,8,10	0	x		0	x		2
P2	13	x	x					3
P3	3,7			x	x			3
* P4	2,7			0	x			3
P5	10,14					x	x	5
P6	7,15			x			x	3
P7	14,15					x	x	3

Tabla 2.4. Tabla de selección de primos implicantes

Examinando cada una de las columnas de mintérminos de la tabla 2.4 se observa que los mintérmino 5, 0 son cubiertos por un solo primo implicante, cada mintérmino que se encuentre en esta situación se encierra en un círculo y se indica con un "\*" que es un primo implicante esencial.

Al elegirlos como primos implicantes se han cubierto a los mintérminos 5,7 y 0,2,8,10; el problema ahora consiste en elegir a los primos implicantes no esenciales que cubran a los mintérminos 3,14 y 15.

El cuarto y último paso es crear una nueva tabla 2.5. de primos implicantes reducida con los mintérminos que no han sido cubiertos y primos implicantes restantes.

Debido a que todos los primos implicantes restantes tienen el mismo costo, se elige a aquellos que cubran a los mintérminos 3,14,15. El primo implicante P7 cubre al 14 y 15 y se puede incluir en la función mínima a P2 o P3 porque ambos cubren al mintérmino 3. La mínima expresión es:

## II.2. ALTERNATIVAS DE SOLUCION

$$F(A,B,C,D) = P1+P2+P6+P7 = B'D'+A'B'C+A'BD+ABC$$

$$F(A,B,C,D) = P1+P3+P6+P7 = B'D'+A'CD+A'BD+ABC$$

Cualquiera de las dos expresiones es mínima porque tienen el mismo costo.

Ecuación	Termino Producto	Minterminos			Costo
		3	4	5	
P1	2,3	X			3
P2	3,7	X			3
P3	10,14		X		3
P6	7,15		X	X	5
P7	14,15	X	X		5

Tabla 2.5. Tabla de implicantes primos reducida

La selección de primos implicantes se realiza analizando cual de ellos tiene el menor costo y que cubra el mayor número de minterminos. Este análisis tiende a dificultarse conforme aumenten los primos implicantes de una función; por tal motivo este proceso de selección debe de realizarse por medio de un algoritmo que facilite la elección.

El algoritmo que se va utilizar esta basado en las siguientes definiciones:

Una columna es cualquier mintermino que se encuentra en una tabla de primos implicantes.

Un renglón es un primo implicante que esta formado de una o varias columnas.

Dos renglones P1 y P2 de una tabla de primos implicantes que tienen una X en las mismas columnas son iguales (P1 = P2).

Dadas dos columnas i y j que forman una tabla de primos implicantes. La columna i domina a la columna j (i > j), si i=j o si la columna i tiene x en todos los renglones que la columna j. La columna i es dominante y j dominada.

El renglón P1 domina al renglón P2 (P1 > P2) si P1 = P2 o si el renglón P1 tiene X en todas las columnas que el renglón P2. El renglón P1 es dominante y P2 dominado.

Dada la característica de que un renglón y una columna pueden ser dominantes o dominados, se crean las dos siguientes reglas:

Regla 1: Un renglón que es dominado por otro renglón es removido de la tabla sin afectar la solución de la función mínima,

siempre y cuando el costo del renglón dominado sea mayor que el del dominador.

**Regla 2 :** Una columna que domina a otra columna es removida de la tabla sin afectar la solución de la función mínima.

Una tabla de primos implicantes es semicíclica si no tiene primos implicantes esenciales; es decir, en cada columna existe por lo menos dos X o no existe ninguna relación dominante entre renglones y columnas, y el costo de los renglones es el mismo. Cuando el costo es igual en todos los renglones, la tabla semicíclica se considera cíclica.

Para resolver una tabla de primos implicantes semicíclica, se debe seleccionar un renglón que tenga el menor costo para incluirlo en la mínima suma y poder usar las técnicas de reducción para remover renglones y columnas de la tabla. Todo este proceso debe ser repetido para cada renglón que puede reemplazar al renglón original y obtener una mínima suma comparando los costos de las expresiones que resultan de la selección arbitraria de un renglón. Para una tabla cíclica la elección puede ser para cualquier renglón y no es necesario comparar diferentes expresiones debido a que tienen el mismo costo.

El algoritmo para encontrar la mínima expresión de una función es (IRWBO):

1) Identificar primos implicantes esenciales, si existen en la tabla original. Localizar los primos implicantes esenciales secundarios en la tabla reducida si cualquier mintermino es cubierto por un solo primo implicante.

2) Remover renglones correspondientes para identificar los primos implicantes esenciales y esenciales secundarios. Remover columnas correspondientes a los minterminos cubiertos por los renglones removidos.

3) Si resulta una tabla semicíclica o cíclica después de completar el paso 2, ejecutar el paso 5. De otra forma aplique el procedimiento de reducción regla 1 y 2.

4) Si una tabla semicíclica o cíclica resulta después del paso 3, realizar el paso 5. De otra forma retornar al paso 1.

5) Aplicar el procedimiento para una tabla semicíclica. Repetir el paso 5 hasta que se produzca una tabla no cíclica. De otra forma retornar al paso 1.

El procedimiento termina cuando el paso 2 y 5 no generan una tabla.

## 11.2. ALTERNATIVAS DE SOLUCION

Cada uno de los primos implicantes esenciales son identificados en la tabla de primos implicantes por un asterisco "\*". Los primos implicantes esenciales secundarios son identificados por un doble asterisco "\*\*".

Por medio de los siguientes ejemplos se explica con detalle el proceso de selección de primos implicantes esenciales y primos implicantes.

Utilizando la tabla 2.3 de primos implicantes se genera la tabla 2.6. En el paso 1 es reconocido P4 y P1 como primos implicantes esenciales, siendo marcados con un asterisco. En el paso 2 se elimina de la tabla a P4 y los minterminos 5 y 7, a P1 y los minterminos 0,2,8, y 10.

Etiqueta	Termino Producto	Minterminos						Costo					
		0	2	3	5	7	8		10	14	15		
* P1	0,2,8,10	*						*					2
P2	2,3			x									3
P7	3,7			x									3
* P4	5,7				*								3
P5	10,14					x			x				3
P6	7,15									x			3
P7	14,15								x	x			3

Tabla 2.6. Tabla de selección de primos implicantes

Para el paso 3, P3 es cubierto por P2, P5 es cubierto por P7. P6 es cubierto por P7. Así que P2, P5, P6 son removidos de la tabla 2.7. El primo implicante dominante tiene un costo menor que cualquier otro. La columna 14 es dominada por la columna 15, por lo tanto se elimina la columna 14.

Etiqueta	Termino Producto	Minterminos		Costo
		3	14 15	
* P3	3,7	*		1
* P7	14,15	*	*	2

Tabla 2.7. Tabla de primos implicantes reducida

El paso 4 nos sirve para verificar si la tabla es cíclica. En este caso no es cíclica y saltamos al paso 1.

Paso 1. Son identificados P3 y P7 como primos implicantes esenciales secundarios y son marcados con doble asterisco.

Paso 2. Son removidos P3 y P7 y las columnas 3 y 14.

Obteniendo la siguiente expresión final:

$$F(A,B,C,D) = P1+P3+P6+P7, \text{ entonces}$$

$$F(A,B,C,D) = B'D'+A'B'C+A'BD+ABC.$$

Funciones que contienen términos no importa.

El método de Quine McCluskey se puede aplicar en funciones que contienen términos no importa, con una modificación en el proceso para obtener la nueva función.

Todos los términos no importa son incluidos en la lista general (Tabla 2.3) para obtener cada uno de los primos implicantes que forman la función. Al momento de crear la tabla de primos implicantes no son incluidos, porque solamente se necesita cubrir a los mintérminos que hacen a la función 1.

Ejemplo 2. Minimize la siguiente función que incluye términos no importa.

$$F(A,B,C,D) = E_m(1,4,7,10,13) + E_d(5,14,15)$$

En el listado 3 de la tabla 2.8 tenemos dos mintérminos generados en diferentes agrupaciones que son idénticos, por lo tanto, se elimina uno de ellos sin algún riesgo de error.

Listado1		Listado2		Listado3	
Mintérminos	ABCD	Mintérminos	ABCD	Mintérminos	ABCD
1	0010	✓	1,5	0,01	P2 5,7,13,15
4	1000	✓	4,5	010_	
5	0011	✓	5,7	01,1	✓
10	0101	✓	5,13	_101	
7	1010	✓	10,14	1_10	P4
13	0111	✓	7,15	_111	
14	1110	✓	13,15	11_1	✓
15	1111	✓	14,15	111_	

Tabla 2.8. Listados de primos implicantes

La tabla 2.9 se forma sólo con los mintérminos de la función y se muestra a continuación:

Etiqueta	Termino Producto	Mintérminos					Costo
		1	4	7	10	13	
** P1	17,13,15						2
** P2	15						3
** P3	4,5						3
** P4	10,14						3
FS	14,15						?

Tabla 2.9. Tabla de selección de primos implicantes esenciales

Paso 1. Se seleccionan como primos implicantes esenciales a P1, P2, P3 y P4.

Paso 2. Son eliminados P1, P2, P3 y P4; las columnas 1, 4, 7, 10 y 13.

Al no generarse una nueva tabla la función mínima es:

$$F(A,B,C,D) = P1+P2+P3+P4$$

$$F(A,B,C,D) = BD+A'C'D+A'BC'+ACD'$$

#### Funciones con salidas múltiples.

Algunos sistemas trabajan con un conjunto de salidas que dependen de  $n$  variables de entrada comunes para efectuar una función. Para este tipo de sistemas Quine McCluskey utiliza dos criterios que se deben aplicar en cada una de las funciones de salida que fueron minimizadas.

a) Cada función minimizada debe tener tantos términos en común como sea posible.

b) Cada función minimizada debe tener un mínimo número de términos producto con el menor de variables.

El algoritmo utilizado en la obtención de la expresión mínima de una función, se aplica para cualquier número de funciones con algunas variantes. El costo del circuito debe ser menor comparado con el que se obtiene si se minimiza individualmente.

Las variantes que hay que considerar son:

1) Cada una de las funciones debe tener una etiqueta para identificar a que mintérmino pertenece; esto es con el fin de obtener los primos implicantes de todas las funciones simultáneamente. En la

## 11.2. ALTERNATIVAS DE SOLUCION

tabla de búsqueda de primos implicantes y en la tabla de primos implicantes se anexa una columna de etiquetas.

2) Dos mintérminos se pueden combinar si tienen una o más etiquetas en común.

3) Cada mintérmino puede ser marcado (✓) si y solo si el nuevo término tiene una etiqueta que es idéntica a alguna de las que poseen los mintérminos.

4) Se considera a una función semicíclica, si lo es con respecto a cada una de las funciones de salida. El criterio de selección para un renglón e incluirlo en la mínima expresión, es que posea el más bajo costo y que se encuentre en varias funciones de salida.

Ejemplo 3: Encontrar la mínima solución en suma de productos de las siguientes funciones.

$$F1(A,B,C,D) = \Sigma m(1,3,4,5,6,7,10,12,13) + \Sigma d(2,9,15)$$

$$F2(A,B,C,D) = \Sigma m(2,3,4,7,9,11,12,13,14) + \Sigma d(1,10,15)$$

$$F3(A,B,C,D) = \Sigma m(0,6,9,10,13) + \Sigma d(1,3,8)$$

La obtención de primos implicantes se muestra en la tabla 2.10. El término 0,8 se forma al combinar el mintérmino 0 y 8, se observa que ambos poseen la misma etiqueta siendo ambos marcados con una (✓).

El mintérmino 2 tiene la etiqueta 110 y el mintérmino 6 la 101, debido a que son comunes en la función F1, la etiqueta del término 2,6 debe ser 100, los mintérminos 2 y 6 no son marcados. Al combinarse todos los mintérminos, se genera la tabla 2.11.

A continuación se lleva a la práctica el algoritmo para minimizar funciones.

Paso 1. Se elige como primo implicante esencial para la función F2 el primo P18, para la función F3 a P1, P21 y P25.

Paso 2. Son removidas las columnas 4,12 de la función F2 y las columnas 0,6,9,13 de la función F3. El único renglón que es removido de la tabla es P1. Los primos implicantes P18, P21 y P25 se conservan para ser eliminados posteriormente por alguna otra técnica de reducción.

Paso 3. No existen columnas dominantes. Los renglones dominados son P23, P25 y P18. P20 domina a P23, P5 domina a P25, P6 domina a P18, por lo que son eliminados P18, P23 y P25.

Paso 4. Se realiza un salto al Paso 1.

Paso 1. No existen primos implicantes esenciales.

Paso 2. No se lleva a cabo.

Paso 3. Existen columnas dominantes. La columna 5 domina a la 4, la 7 domina a la 6, la 13 a la 12 y la 3 a la 7. Las columnas dominantes son eliminadas (3,5,7,13). Los renglones dominantes son P11 que domina a P21, P24 y P3, y P2 domina a P4 y P14. Los renglones dominados son eliminados (P3,P4,P14,P21,P24).

Paso 4. Se realiza un salto al Paso 1.

Paso 1. No existen primos implicantes esenciales.

Paso 2. No se lleva a cabo.

Paso 3. No existen columnas ni renglones dominantes.

Paso 4. La función es semicíclica, por lo tanto, se efectúa el paso 5.

Paso 5. Se elige como primo implicante esencial secundario a P13 para F2 porque tiene un costo menor y cubre a varias. Las columnas que se eliminan son 13 y 14. La nueva tabla ya no es semicíclica y se efectúa el paso 1.

Paso 1. No existen primos implicantes esenciales secundarios.

Paso 2. No se lleva a cabo.

Paso 3. No existen columnas dominantes. Existen renglones dominantes. El renglon PB domina a P22 y P6 domina a P12. Los renglones eliminados son P22 y P12.

Paso 4. Se realiza un salto al Paso 1.

Paso 1. Se elige como primo implicante esencial secundario a PB de F1.

Paso 2. Se elimina la columna 4 y 12.

Paso 3. No existen columnas dominantes. Existe un renglon dominante. El renglon P5 domina a P7.

Paso 4. Se realiza un salto al Paso 1.

Se efectúa el mismo procedimiento hasta obtener la tabla 2.12. Donde se observa que P2 es primo implicante esencial para F1 y



P19 para F3. P6 es primo implicante esencial secundario para F2 y P17 para F1. Las funciones minimizadas son:

$$F1(A,B,C,D) = BC' + A'C + A'D + B'CD'$$

$$F2(A,B,C,D) = BC'D' + AB + CD + AD + B'C$$

$$F3(A,B,C,D) = B'C' + A'BCD' + AC'D + AB'D'$$

Este método es utilizado para funciones que esten en producto de sumas y el procedimiento tiene dos cambios. El primero, se refiere a que en la tabla de determinación de primos implicantes se cambia la lista de mintérminos por maxtérminos. El segundo, es al finalizar el método donde los términos productos se complementan para formar los términos suma.

Listado 1					Listado 2					Listado 3				
Mintérminos	ABCD	F1	F2	F3	Mintérminos	ABCD	F1	F2	F3	Mintérminos	ABCD	F1	F2	Pi
0	0000	0	0	1	0,1	000	0	0	1	0,1,8,9	0	0	1	P1
1	0001	1	1	1	0,8	000	0	0	1	1,3,5,7	0	1	0	P2
2	0010	1	1	0	1,3	00	1	1	1	1,3,9,11	0	1	0	P3
4	0100	1	1	0	1,5	0	0	1	1	1,5,9,13	0	1	0	P4
8	1000	0	0	1	1,9	00	1	1	1	2,3,6,7	0	1	0	P5
3	0011	1	1	1	2,3	00	1	1	0	2,3,10,11	0	1	0	P6
5	0101	1	0	0	2,6	0	1	0	0	4,5,6,7	0	1	0	P7
6	0110	1	0	1	2,10	0	1	0	0	4,5,12,13	1	0	0	P8
9	1001	1	1	1	4,5	0	1	0	0	3,7,11,15	1	1	0	P9
10	1010	1	1	1	4,6	0	1	0	0	5,7,13,15	1	1	0	P10
12	1100	1	1	0	4,12	0	1	0	0	9,11,13,15	1	1	0	P11
7	0111	1	1	0	6,9	1	0	0	1	10,11,14,15	1	1	0	P12
11	1011	0	1	0	8,10	1	0	0	1	12,13,14,15	1	1	0	P13
13	1101	1	1	1	3,7	0	1	1	0	3,11	0	1	0	P20
14	1110	0	1	0	3,11	0	1	1	0	5,7	0	1	0	P21
15	1111	1	1	0	5,13	0	1	0	0	5,13	0	1	0	P22
					6,7	0	1	0	0	6,7	0	1	0	P23
					9,11	1	0	1	0	9,11	1	0	1	P24
					9,13	1	0	1	1	10,11	1	0	1	P25
					10,11	1	0	1	0	10,14	1	0	1	P26
					10,14	1	0	1	0	12,13	1	0	1	P27
					12,13	1	0	1	0	12,14	1	0	1	P28
					12,14	1	0	0	1	7,15	1	1	0	P29
					7,15	1	1	1	0	11,15	1	1	0	P30
					11,15	1	1	0	1	13,15	1	1	0	P31
					13,15	1	1	1	0	14,15	1	1	0	P32
					14,15	1	1	0	1					

Tabla 2.10. Lista de primos implicantes





II.3. JUSTIFICACION DE LA ALTERNATIVA A SELECCIONAR.

Al comparar los métodos descritos se tienen las ventajas y desventajas que se describen a continuación:

El método de álgebra de Boole cumple con la función de simplificar las funciones, sin embargo, no es sencillo de aplicarse debido a que no existe un procedimiento específico que utilice los teoremas y axiomas para elegir el camino correcto de minimización. Esto constituye una desventaja considerable como alternativa.

El método de Mapa de Karnaugh tiene la ventaja de poseer un procedimiento para efectuar la minimización más directa y rápida que los otros dos métodos. Sin embargo, éste método se torna difícil para una función que presenta más de cinco variables, pues no se tiene seguridad de que el resultado sea mínimo. El realizar una selección mínima depende de la habilidad del diseñador, ya que el aumento considerable de celdas dificulta una elección acertada.

El método Quine McCluskey tiene la ventaja de ser un procedimiento que garantiza la minimización, además es aplicable a funciones que cuentan con muchas variables. Debido a que éste método tiene una secuencia de pasos bien definidos, existe la posibilidad de ser programado.

Para un diseñador que realiza manualmente el método representa un proceso monótono, por lo tanto, existe la probabilidad de cometerse errores. Debido a sus ventajas, el método Quine McCluskey, es el método seleccionado para desarrollar el "Sistema de Minimización".

Se tiene como antecedente la existencia de un sistema que maneja cartas ASM como datos de entrada, siendo minimizada por el método Quine McCluskey y programando a un PLA (Arreglo lógico programable), que representa cada una de las salidas del circuito lógico. Este sistema se desarrolló en Posgrado de Ingeniería.

El desarrollar este sistema permite al diseñador de sistemas digitales obtener una expresión mínima en el menor tiempo posible, con la seguridad de que no existen errores humanos.

Para el estudiante de Ingeniería sirve de apoyo porque podrá reafirmar algunos de los conocimientos que va adquiriendo en el área de sistemas digitales.

**ANALISIS DEL SISTEMA**

El análisis de requerimientos es la parte fundamental del desarrollo de un sistema debido a que su éxito depende de cada una de las especificaciones que se describan y de la forma en que se detallen.

Cada una de las técnicas de análisis que existen están relacionadas por un conjunto de principios fundamentales (PREBB).

1. El dominio de la información así como el dominio funcional de un problema debe ser representado y comprendido.
2. El problema debe subdividirse de forma que se descubran los detalles de una manera progresiva (o jerárquica).

Aplicando estos principios se encamina el dominio de la información de forma que pueda comprenderse su función auxiliado por diagramas de flujo de datos y un diccionario de datos. La partición se aplica para reducir la complejidad del problema estableciendo interfaces entre las partes.

La técnica utilizada para el análisis de requerimientos es la de análisis estructurado de sistemas, en la que existen cuatro elementos básicos, diagramas de flujo de datos, diccionario de datos, representaciones lógicas de procedimientos y estructuración de almacenamiento de datos (RICB7).

Las representaciones lógicas del procesamiento se utilizan para especificar la secuencia de procesos. Es mejor que éstas junto con la estructuración de almacenamiento sean diferidas hasta la fase de diseño detallado.

**III.1. DEFINICION DEL SISTEMA.**

El "Sistema de Minimización" es creado para servir al usuario como una herramienta que realice la minimización de funciones de Boole, y la visualización del diagrama lógico correspondiente.

Las metas a alcanzar por el Sistema son :

- Obtener un resultado confiable, equivalente y minimizado de la función introducida.
- Obtener el diagrama lógico de la función minimizada, el cual es desplegado en pantalla, en modo gráfico.
- Obtener el diagrama lógico impreso de la función minimizada.

- Tiempo de respuesta óptimo.
- Uso de menús encadenados.
- La graficación permite movimientos verticales y horizontales para poder observar el diagrama lógico en pantalla.

El usuario del sistema podrá ser cualquier persona familiarizada con el área digital y podrá usarlo en una microcomputadora IBM personal o compatible.

### III.2. REQUERIMIENTOS DEL SISTEMA.

#### Resumen y panorama del Sistema.

El propósito de este sistema es automatizar el método Quine McCluskey para minimizar funciones de Boole y debe tener la capacidad de procesar una o varias funciones. Esto es importante, debido a que se pueden obtener ahorros considerables cuando se comparten términos entre las funciones.

Cada función a minimizar puede tener hasta 15 variables permitiendo trabajar con 32,768 términos diferentes. El sistema maneja 15 funciones.

Dado el volumen de datos que son procesados durante la captura de datos, minimización, graficación e impresión es necesario crear y utilizar archivos en disco.

Como resultado de las necesidades que como usuario se tienen con este sistema, los requerimientos son los siguientes:

- a) Entrada de datos. Los mintérminos o maxtérminos son introducidos a través del teclado o por un archivo previamente salvado.
- b) Minimización. Que muestre el resultado del proceso de minimización de la(s) función(es) introducida(s).
- c) Graficación. Que grafique el diagrama lógico de una o de todas las funciones previamente minimizadas, en pantalla.
- d) Impresión. Que imprima el diagrama lógico de una o de todas las funciones previamente minimizadas y la función de Boole minimizada.
- e) Opciones. Que permita editar (Eliminar o Insertar) las funciones que han sido capturadas.

- f) Salir. Finalizar la sesión de trabajo cuestionando si se está seguro de terminar.
- g) Ayuda en línea. Que cuente con una pequeña descripción de las teclas más importantes del sistema en pantalla.
- h) Ayuda del sistema. Que el sistema apoye al usuario describiendo las actividades que realiza cada opción. El usuario debe de estar posicionado en la misma.
- i) Manejo de errores. Que el sistema proporcione mensajes de aviso o de error de acuerdo a la interrupción presentada en la ejecución de algún proceso.

La figura 3.1. muestra las funciones del "Sistema de Minimización".

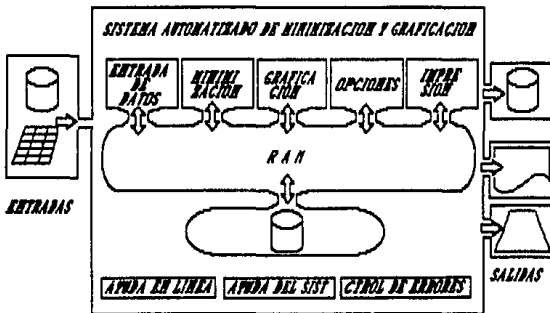


Figura 3.1. Funciones del "Sistema de Minimización".

Ambientes de desarrollo, operación y mantenimiento.

Los recursos computacionales con que se cuenta para desarrollar el Sistema son:

- Computadora PC.
- Monitor a color o video compuesto.
- Software para el procesamiento de texto (Documentación).



### III.2. REQUERIMIENTOS DEL SISTEMA

- Software para publicaciones gráficas (Documentación).
- Lenguaje de Programación C.
- Impresora matriz de puntos.

#### Flujo de datos.

Los diagramas de flujo de datos representan la forma en la que la información fluye a través del sistema. Describen las fuentes y destinos de los datos, así como sus transformaciones y almacenamientos. El diagrama se expone con un nivel de detalle simple y es presentado en la figura 3.2 de una manera formal.

Las fuentes y destinos de los datos son presentados con figuras propias del dispositivo a emplear, las transformaciones de los datos se presentan con un rectángulo y el almacenamiento por rectángulos sin el extremo derecho y el flujo de información mediante flechas que son etiquetadas con nombres de datos cuyas características se especifican en el diccionario de datos.

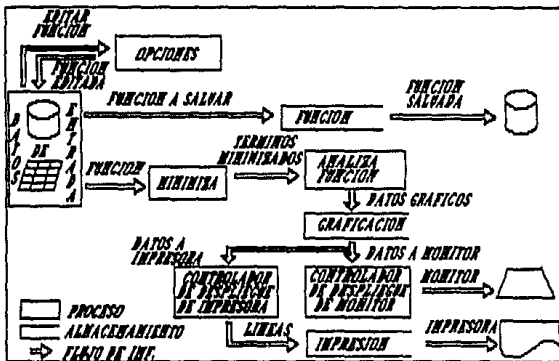


Figura 3.2. Diagrama formal del flujo de datos.

#### Diccionario de datos.

NOMBRE : Función

PROPOSITO : Entrada de datos al proceso por medio de teclado o por un archivo en disco para ser minimizado

### III.2. REQUERIMIENTOS DEL SISTEMA

- DERIVADO DE :** Datos de entrada  
**SUBELEMENTOS :** Número de función  
                  Término (Valor decimal)  
                  Tipo de función  
                  Términos no importa
- NOMBRE :** Función a salvar  
**PROPOSITO :** Entrada de datos al proceso por medio de teclado o por un archivo en disco para ser grabados en disco
- DERIVADO DE :** Datos de entrada  
**SUBELEMENTOS :** Número de función  
                  Término (Valor decimal)  
                  Tipo de función  
                  Términos no importa
- NOMBRE :** Función salvada  
**PROPOSITO :** Grabar función previamente introducida
- DERIVADO DE :** Datos de entrada  
**SUBELEMENTOS :** Número de función  
                  Término (Valor decimal)  
                  Tipo de función  
                  Términos no importa
- NOMBRE :** Términos minimizados  
**PROPOSITO :** Determinar las compuertas y líneas empleando los términos de la función minimizada
- DERIVADO DE :** Minimiza  
**SUBELEMENTOS :** Número de función  
                  Término (Valor decimal)  
                  Tipo de función
- NOMBRE :** Datos gráficos  
**PROPOSITO :** Almacenar los datos gráficos de la función
- DERIVADO DE :** Analiza función  
**SUBELEMENTOS :** Tipo de compuerta  
                  Número de entradas  
                  Número de función  
                  Coordenadas de pantalla
- NOMBRE :** Datos a monitor  
**PROPOSITO :** Obtener los datos que requiere el controlador de despliegues del monitor para graficar en pantalla el circuito lógico de la función minimizada
- DERIVADO DE :** Graficación  
**SUBELEMENTOS :** Tipo de compuerta  
                  Número de entradas  
                  Número de función  
                  Coordenadas de pantalla

### III.2. REQUERIMIENTOS DEL SISTEMA

- NOMBRE :** Datos a impresora  
**PROPOSITO :** Obtener los datos que requiere el controlador de despliegues de impresora para imprimir el circuito lógico
- DERIVADO DE :** Graficación  
**SUBELEMENTOS :** Tipo de compuerta  
Número de entradas  
Número de función
- NOMBRE :** Líneas  
**PROPOSITO :** Almacenar temporalmente cada línea gráfica de impresión para ser enviada a la impresora
- DERIVADO DE :** Controlador de despliegue de impresora
- NOMBRE :** Diagrama a monitor  
**PROPOSITO :** Dibuja en pantalla el circuito lógico correspondiente a la función minimizada
- DERIVADO DE :** Controlador de despliegue del monitor
- NOMBRE :** Diagrama a impresión  
**PROPOSITO :** Imprime en la hoja de impresión el circuito lógico correspondiente a la función minimizada
- DERIVADO DE :** Impresión

#### Manejo de excepciones.

La Tabla 3.1 contiene las condiciones para las cuales el procesamiento ya no procede de manera normal, teniendo como respuesta un mensaje que describe el motivo por el cual se interrumpe.

CONDICION DE EXCEPCION	MENSAJE	NO. ERROR
Introduce términos no impuestos, sin existir mintérminos.	No existen mintérminos en la función.	1
Introduce términos no impuestos, sin existir maxtérminos.	No existen maxtérminos en la función.	2
Captura caracteres.	Sólo valores numéricos.	3
Captura funciones o variables menores o iguales a cero.	El dato debe ser mayor que cero.	4
Mintérmino capturado previamente. función.	El mintérmino ya existe en la función.	5
Maxtérmino capturado previamente.	El maxtérmino ya existe en la función.	6

CONDICION DE EXCEPCION	MENSAJE	NO. ERROR
El <u>mintérmino</u> sobrepasa el límite de entrada.	Mintérmino fuera del rango de entrada.	7
El <u>maxtérmino</u> sobrepasa el límite de entrada.	Maxtérmino fuera del rango de entrada.	8
Número de variables mayor al límite.	El número máximo de variables es 15.	9
Número de funciones mayor al límite.	El número máximo de funciones es 15.	10
Opción de minimización sin existir funciones de entrada.	No existen funciones a minimizar.	11
Captura de funciones sin <u>min</u> términos o <u>max</u> términos.	Funciones incompletas : Datos eliminados.	12
Selección de función <u>inexistente</u> para ser editada.	No existe la función.	13
Error en la captura de <u>término</u> o término no importa en Opciones.	Pulse sólo 0 o 1.	14
Elimina un <u>mintérmino</u> que no existe.	El <u>mintérmino</u> no existe en la función.	15
Elimina un <u>maxtérmino</u> que no existe.	El <u>maxtérmino</u> no existe en la función.	16
Selección de Salva sin <u>existir</u> datos en memoria.	No existen funciones en memoria.	17
El archivo no tiene la <u>estructura</u> lógica necesaria.	Error de lectura en el archivo.	18
Selección de Opciones sin existir datos en memoria.	No existen elementos en la <u>lista</u> .	19
Selección de Graficación o Impresión sin existir <u>funciones</u> minimizadas.	No existen funciones minimizadas.	20
Impresora sin hojas.	La impresora no tiene hojas.	21
Impresora fuera de línea.	La impresora está fuera de línea < Continuo > S/N:	22

## III.2. REQUERIMIENTOS DEL SISTEMA

CONDICION DE EXCEPCION	MENSAJE	NO. ERROR
Puerto desconectado.	El puerto está desconectado < Continuo > S/N:	23
Impresora apagada.	La impresora está apagada <Con- tinuo> S/N:	24
Nombre de archivo reservado.	Nombre de archivo reservado.	25
No existen todos los datos necesarios para cargar el archivo.	Archivo incompleto.	26
No existe el archivo de ayuda (Ayuda.ayu).	No existe el archivo de ayuda.	27
Disco de trabajo lleno.	Insuficiente espacio en disco.	28

### 3.1 Tabla de excepciones con mensaje de error.

La tabla 3.2 contiene las condiciones para los cuales el procesamiento se interrumpe momentáneamente.

CONDICION DE EXCEPCION	MENSAJE DE AVISO
Proceso de captura interrumpido.	¿Quiere continuar con el proceso de lectura S/N : ?
Captura de mintérminos terminada.	¿Son todos los mintérminos a insertar S/N:?
Captura de maxtérminos terminada.	¿Son todos los maxtérminos a insertar S/N:?
Nueva captura de funciones.	Nuevas funciones S/N:
El usuario selecciona el proceso de impresión.	La impresora está lista S/N :
Afectar un proceso inmediatamente.	Pulse < ESC > para abortar
El usuario da fin a su sesión de trabajo.	Esta usted seguro de salir del sistema S/N :

CONDICION DE EXCEPCION

MENSAJE DE AVISO

Selección de Carga de archi  
vos sin que exista alguno.

No existen archivos en el disco

Tabla 3.2 de excepciones con mensaje de aviso.

**DISEÑO DEL SISTEMA**

Durante el diseño se desarrolla, revisa y documenta las estructuras de datos, la estructura del programa y los detalles procedimentales. Existen una serie de conceptos que son fundamentales para el diseño (PREBB) :

- a) Estructura del programa. Representa la organización (jerárquica) de los componentes del programa (módulos) e implica una jerarquía de control.
- b) Estructura de datos. Es una representación de la relación lógica entre elementos individuales de datos. La estructura de datos dicta la organización, grado de asociatividad y alternativas de procesamiento para la información.
- c) Modularidad. El programa se divide en elementos con nombres y direcciones separadas, llamadas módulos, que se integran para satisfacer los requerimientos del problema.

El diseño de sistemas comprende tres actividades: diseño de datos, diseño arquitectónico y diseño detallado.

El diseño de datos tiene como actividad primaria seleccionar las representaciones lógicas de los objetos de datos. La selección puede implicar análisis algorítmicos de estructuras alternativas o identificar módulos de programas que deben operar directamente sobre las estructuras de datos lógicas.

El diseño arquitectónico tiene como objetivo principal desarrollar una estructura de programa modular y representar las relaciones de control entre los módulos. Mezcla la estructura de programas y estructura de datos definiendo interfaces que facilitan el flujo de datos a lo largo del programa.

El diseño detallado se realiza después de establecer la estructura del programa y de los datos. Se detalla cada uno de los procedimientos utilizando diagramas de flujo o pseudocódigo.

**IV.1. ORGANIZACION DE LOS DATOS.**

Las estructuras de datos que se usan son las siguientes:

- a) Estructura LIST\_FUNC\_TAB. Esta estructura es empleada para almacenar todas las características de los términos capturados y

los primos implicantes determinados en el proceso. Sus componentes son los siguientes :

NOMBRE	NEMONICO	TIPO
Apuntador a la estruct. anterior	pizq	apunt. estructura
Valor del término	val_dec	apunt. caracter
Cantidad de unos o potencia	num_uno_pot	apunt. caracter
Funciones que lo incluyen	num_func	apunt. caracter
Marca de combinación	marca	apunt. caracter
Bandera de término o término no importa	mint_noimp	apunt. caracter
Número de primo implicante	num_primo	apunt. caracter
Apuntador a la estruct. siguiente	pder	apunt. estructura

b) Estructura TERM\_MINT\_MAXT. Esta estructura es empleada para almacenar las tablas y tablas reducidas de los implicantes primos esenciales y no esenciales. Posteriormente almacena la función mínima. Sus componentes son los siguientes :

NOMBRE	NEMONICO	TIPO
Apuntador a la estruct. anterior	pizq	apunt. estructura
Términos	mint_maxt	apunt. caracter
Primos implicantes	num_uno_pot	apunt. caracter
Apuntador a la estruct. siguiente	pder	apunt. estructura

c) Estructura NOMBRES. Esta estructura es empleada para almacenar los nombres de los archivos con extensión bol. De estos nombres el usuario seleccionará uno para ser cargado. Sus componentes son los siguientes:

NOMBRE	NEMONICO	TIPO
Apuntador a la estruct. anterior	pizq	apunt. estructura
Nombre del archivo	arch	apunt. caracter
Apuntador a la estruct. siguiente	pder	apunt. estructura

d) Estructura de unión CIRCUITO. Esta estructura es la unión de las estructuras Línea y Compuerta que son empleadas para almacenar las características de cada línea o compuerta del diagrama lógico de las funciones minimizadas. Los componentes de las estructuras son los siguientes :



#### IV.1. ORGANIZACION DE LOS DATOS

##### Estructura de Línea.

NOMBRE	NEMONICO	TIPO
Coordenada en X1 de línea	CoordX1	entero
Coordenada en Y1 de línea	CoordY1	entero
Coordenada en X2 de línea	CoordX2	entero
Coordenada en Y2 de línea	CoordY2	entero
Color de la línea	Color	caracter
Variable de la línea	NomVariable	caracter

##### Estructura de Compuerta.

NOMBRE	NEMONICO	TIPO
Coordenada en X de la compuerta	CoordX1	entero
Coordenada en Y de la compuerta	CoordY1	entero
Número de entradas que contiene	NumEntradas	caracter
Color de la compuerta	Color	caracter
Selección del tipo de compuerta : OR, AND, inversor.	TipoComp	caracter
Dirección de la compuerta : Arriba, Abajo, Izquierda, Derecha	DirecComp	caracter

Para seleccionar una línea o una compuerta se utiliza el campo Elección y se usan los apuntadores a la estructura de unión :

NOMBRE	NEMONICO	TIPO
Selección de compuerta o línea	Eleccion	caracter
Apuntador a la estruct. anterior	pizq	apunt. a unión
Apuntador a la estruct. siguiente	pder	apunt. a unión

e) Estructura IMPDIAGAUx. Esta estructura es empleada para almacenar temporalmente todas aquéllas figuras que esten situadas en la línea de impresión que es transmitida a la impresora. Sus componente son los siguientes :

NOMBRE	NEMONICO	TIPO
Apuntador a la estruct. anterior	pizq	apunt estructura
Coordenada en X1	CoordX1	entero
Coordenada en Y1	CoordY1	entero
Coordenada en X2	CoordX2	entero

#### IV.1. ORGANIZACION DE LOS DATOS

Coordenada en Y2	CoordY2	entero
Tipo de compuerta	TipoComp	caracter
Elección entre línea o compuerta	Eleccion	caracter
Número de entradas	NumEntradas	caracter
Nombre de la variable	Variable	caracter
Número de la parte de la figura a imprimir.	MasFigCompta	caracter
Apuntador a la estruct. siguiente	pder	apunt. estructura

f) Estructura MENUVENTANA. Esta estructura es empleada para almacenar ventanas y menús que el sistema solicita. Sus componentes son los siguientes :

NOMBRE	NEMONICO	TIPO
Apuntador a la estruct. anterior	pizq	apunt. estructura
Límite inferior en X	MinX	entero
Límite inferior en Y	MinY	entero
Límite superior en X	MaxX	entero
Límite superior en Y	MaxY	entero
Coordenada actual en X	CurX	entero
Coordenada actual en Y	CurY	entero
Dirección de inicio del video a salvar	p	apunt. caracter
Texto del menú	menu	apunt. caracter
Letras de cada opción	teclas	apunt. caracter
Borde de la ventana	borde	caracter sin signo
Cantidad de opciones	num_opc	entero
Bandera de activación	activa	caracter sin signo
Apuntador a la estruct. siguiente	pder	apunt. estructura

#### IV.2. DISEÑO ARQUITECTONICO.

En ésta fase del sistema se describirá su estructura por medio de cartas de estructura que son esquemas de representación para un desarrollo jerárquico de arriba a abajo (FAIB7). La figura 4.1, es una carta de estructura que especifica los procesos generales del "Sistema de Minimización".

Los diagramas 4.2, 4.3, 4.4, 4.5 y 4.6 identifican las subfunciones de las funciones de alto nivel descritas en el 4.1.

Para ofrecer una vista conceptual del sistema de la figura 4.7 a la 4.30 se establecen las relaciones e interconexiones entre las funciones, los datos y el almacenamiento de los mismos. Cada narrativa contiene el nombre del módulo, la descripción de la función de éste, así como sus entradas y salidas.

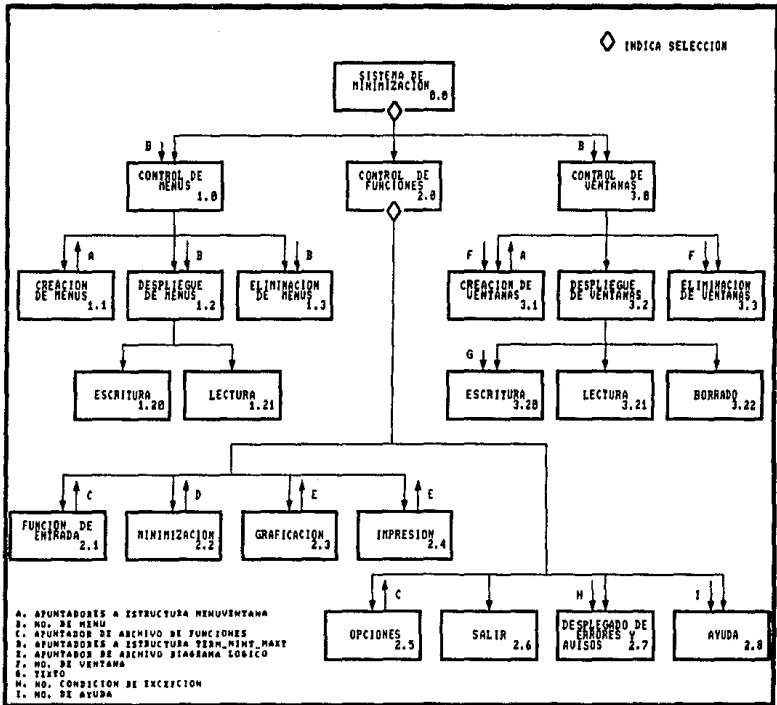


Figura 4.1. Carta estructurada del módulo principal.

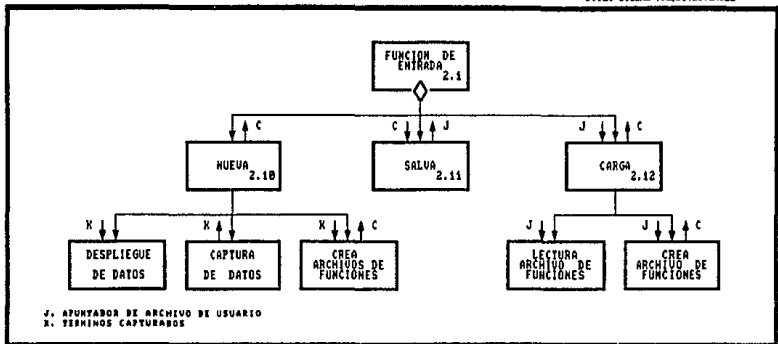


Figura 4.2. Carta estructurada del módulo 2.1.

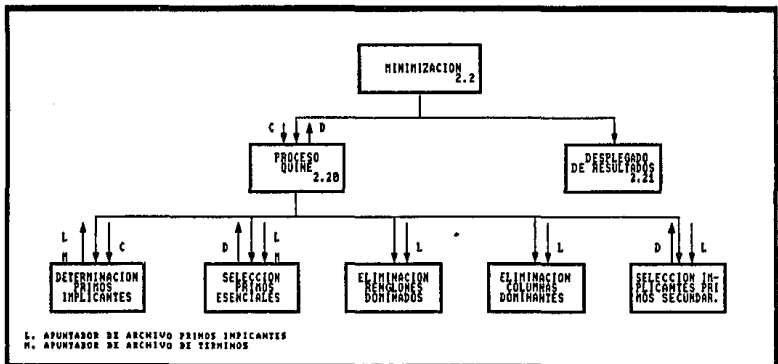


Figura 4.3. Carta estructurada del módulo 2.2.

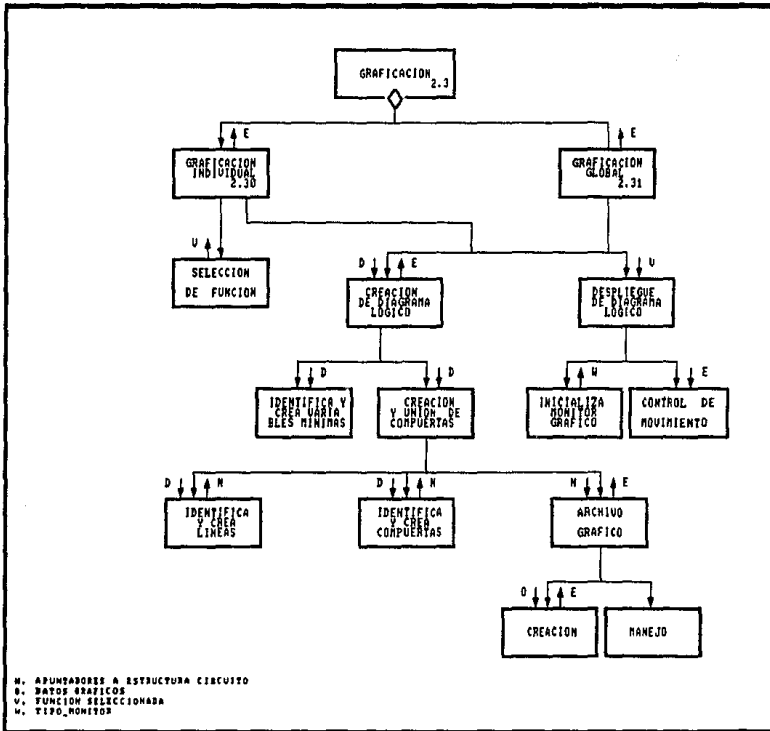


Figura 4.4. Carta estructurada del módulo 2.3.

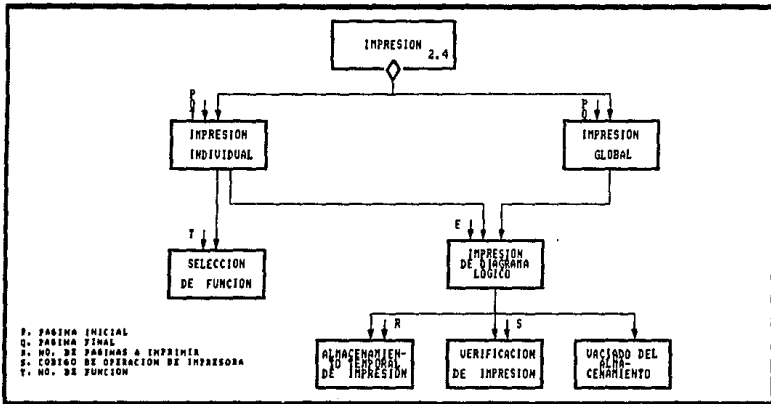


Figura 4.5. Carta estructurada del módulo 2.4.

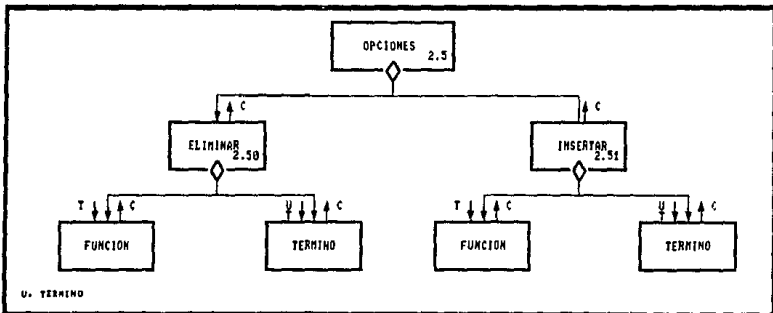


Figura 4.6. Carta estructurada del módulo 2.5.

MÓDULO: SISTEMA DE MINIMIZACIÓN.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
CONTROL DE MENUS	PROPORCIONA AL SISTEMA EL USO DE MENUS ENCADENADOS (PRINCIPALES)	NO. DE MENU	
CONTROL DE FUNCIONES	PROPORCIONA AL SISTEMA EL USO DE FUNCIONES QUE PERMITAN SU DESARROLLO.		
CONTROL DE VENTANAS	PROPORCIONA AL SISTEMA EL USO DE VENTANAS EN LAS QUE PUE- DA DESPLEGARSE LA INFORMACIÓN PROCESADA Y MENSAJES.	NO. DE VENTA- NA	

Figura 4.7. Narrativa del módulo Sistema de Minimización.

MÓDULO: CONTROL DE MENUS.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
CREACION DE MENUS	CREA TODA LA ESTRUCTURA DE LOS MENUS QUE SERAN DESPLEGA- DOS.		OPUNTADORES A ESTRUCTURA REKUEVENTANA
DESPLIEGUE DE MENUS	DESPLIEGA LOS MENUS ENCADENADOS DE ACUERDO A SU SOLICI- TUD.	NO. DE MENU	
ELIMINACION DE MENUS	ELIMINA LOS MENUS QUE FUERON SOLICITADOS EN LA PANTALLA	NO. DE MENU	

Figura 4.8. Narrativa del módulo Control de menús.

MÓDULO: DESPLIEGUE DE MENUS.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCION	ENTRADA	SALIDA
ESCRITURA	DESPLIEGA TODAS LAS OPCIONES DEL MENU SELECCIONADO.		
LECTURA	LEE LAS SOLICITUDES DE DESPLIEGUE.		

Figura 4.9. Narrativa del módulo Despliegue de menús.

MÓDULO: CONTROL DE FUNCIONES.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCION	ENTRADA	SALIDA
FUNCION DE ENTRADA	REALIZA EL PROCESO DE ENTRADA DE DATOS DEL SISTEMA.		APUNTAADOR DE ARCHIVO DE FUNCIONES
MINIMIZACION	REALIZA EL PROCESO DE MINIMIZACION.		APUNTAADOR A ESTRUCTURA TERA_MINI_MAXI
GRAFICACION	REALIZA LA GRAFICACION DEL DIAGRAMA LOGICO DE LAS FUNCIONES MINIMIZADAS.		APUNTAADOR DE ARCHIVO DIAGRAMA LOGICO
IMPRESION	REALIZA LA IMPRESION DE LAS FUNCIONES MINIMIZADAS CON SU CORRESPONDIENTE DIAGRAMA LOGICO.		APUNTAADOR DE ARCHIVO DIAGRAMA LOGICO
OPCIONES	REALIZA LA EDICION DE LA FUNCION DE ENTRADA.		APUNTAADOR DE ARCHIVO DE FUNCIONES
SALIR	PERMITE AL USUARIO SALIR DEL SISTEMA, REGRESANDO AL SISTEMA OPERATIVO.		
DESPLIEGADO DE ERRORES Y AVISOS	PERMITE AL USUARIO CONTAR CON UN DESPLIEGADO DE ERRORES Y AVISOS QUE SE PRESENTAN EN TODA LA OPERACION DEL SISTEMA.	NO. DE CONDICION DE EXCEPCION	
AYUDA	PERMITE AL USUARIO CONTAR CON AYUDA DEL SISTEMA PARA AUXILIARLO EN SU OPERACION.	NO. DE AYUDA	

Figura 4.10. Narrativa del módulo Control de Funciones.



MÓDULO: FUNCIÓN DE ENTRADA.			
NOMBRE	PROCESOS DESCRIPCIÓN	PARÁMETROS	
		ENTRADA	SALIDA
NUEVA	REALIZA LA CAPTURA DE LAS FUNCIONES INTRODUCIDAS.		APUNTAADOR DE ARCHIVO DE FUNCIONES
SALVA	GRABA EN DISCO LAS FUNCIONES DE ENTRADA.	APUNTAADOR DE ARCHIVO DE FUNCIONES	APUNTAADOR DE ARCHIVO DE USUARIO
CARGA	REALIZA LA CARGA DE UN ARCHIVO DE FUNCIONES.	APUNTAADOR DE ARCHIVO DE USUARIO	APUNTAADOR DE ARCHIVO DE FUNCIONES

Figura 4.11. Narrativa del módulo Función de Entrada.

MÓDULO: NUEVA.			
NOMBRE	PROCESOS DESCRIPCIÓN	PARÁMETROS	
		ENTRADA	SALIDA
DESPLIEGUE DE DATOS	DESPLIEGA LOS DATOS QUE SE ESTAN INTRODUCIENDO EN LA VENTANA DE CAPTURA.	TERMINOS CAPTURADOS	
CAPTURA DE DATOS	CAPTURA Y PROCESA LOS DATOS DE LAS FUNCIONES INTRODUCIDAS POR EL USUARIO.		TERMINOS CAPTURADOS
CREA ARCHIVOS DE FUNCIONES	CREA EL ARCHIVO DE LOS DATOS INTRODUCIDOS AL TERMINARSE LA MEMORIA O LA CAPTURA.	TERMINOS CAPTURADOS	APUNTAADOR DE ARCHIVO DE FUNCIONES

Figura 4.12. Narrativa del módulo Nueva.

MÓDULO: CARGA.			
PROCESOS		PARÁMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
LECTURA ARCHIVO DE FUNCIONES	LEE EL ARCHIVO DE FUNCIONES SELECCIONADO POR EL USUARIO	APUNTAADOR DE ARCHIVO DE USUARIO	
CREA ARCHIVO DE FUNCIONES	CREAR ARCHIVOS DE TRABAJO.	APUNTAADOR DE ARCHIVO DE USUARIO	APUNTAADOR DE ARCHIVO DE FUNCIONES

Figura 4.13. Narrativa del módulo Carga.

MÓDULO: MINIMIZACIÓN.			
PROCESOS		PARÁMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
PROCESO QUINE	PROCESA LOS TÉRMINOS PARA DETERMINAR LA FUNCIÓN MINIMA	APUNTAADOR DE ARCHIVO DE FUNCIONES	APUNTAADORES A ESTRUCTURA YEAR_MINI_MAXI
DESPLIEGADO DE RESULTADOS	DESPLIEGA LA FUNCIÓN MINIMIZADA.		

Figura 4.14. Narrativa del módulo Minimización.

## IV.2. DISEÑO ARQUITECTÓNICO

MÓDULO: PROCESO QUINE.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
DETERMINACION PRIMOS IMPLICANTES	DETERMINA LOS PRIMOS IMPLICANTES DE LA FUNCION DE ENTRADA.	APUNTAADOR DE ARCHIVO DE FUNCIONES	APUNT DE ARCH PRIMOS IMPLIC. APUNT DE ARCH TERRINDS
SELECCION PRIMOS ESENCIALES	SELECCIONA LOS PRIMOS ESENCIALES DE LOS PRIMOS IMPLICANTES DETERMINADOS.	APUNT DE ARCH PRIMOS IMPLIC. APUNT DE ARCH TERRINDS	APUNTAADORES A ESTRUCTURA A TERR_MINI_MAXI (INCOMPLETA)
ELIMINACION RENGLONES DOMINADOS	ELIMINACION DE RENGLONES DOMINADOS SI ESTOS EXISTEN.	APUNTAADOR DE ARCHIVO PRIMOS IMPLICANTES (REDUCIDA)	
ELIMINACION COLUMNAS DOMINANTES	ELIMINACION DE COLUMNAS DOMINANTES SI ESTAS EXISTEN.	APUNTAADOR DE ARCHIVO PRIMOS IMPLICANTES (REDUCIDA)	
SELECCION IMPLICANTES PRIMOS SECUNDARIOS	SELECCION DE LOS IMPLICANTES PRIMOS SECUNDARIOS SI ESTOS EXISTEN.	APUNTAADOR DE ARCHIVO PRIMOS IMPLICANTES (REDUCIDA)	APUNTAADORES A ESTRUCTURA A TERR_MINI_MAXI

Figura 4.15. Narrativa del módulo Proceso Quine.

MÓDULO: GRAFICACION.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
GRAFICACION INDIVIDUAL	REALIZA LA GRAFICACION DEL DIAGRAMA LOGICO DE UNA DE LAS FUNCIONES MINORIZADA.		APUNTAADOR DE ARCHIVO DIAGRAMA LOGICO
GRAFICACION GLOBAL	REALIZA LA GRAFICACION DEL DIAGRAMA LOGICO DE TODAS LAS FUNCIONES MINORIZADAS.		APUNTAADOR DE ARCHIVO DIAGRAMA LOGICO

Figura 4.16. Narrativa del módulo Graficación.

MÓDULO: GRAFICACION INDIVIDUAL.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCION	ENTRADA	SALIDA
CREACION DE DIAGRAMA LOGICO	CREA EL DIAGRAMA LOGICO DE LA FUNCION.	APUNTADES A ESTRUCTURA TEAM_MINI_MAXI	APUNTADES A ARCH. DE DIA- GRAMA LOGICO.
SELECCION DE FUNCION	PERMITE AL USUARIO SELECCIONAR UNA DE LAS FUNCIONES A SER GRAFICADA.		FUNCION SELECCIONADA
DESPLIEGUE DE DIAGRAMA LOG.	DESPLIEGA EL DIAGRAMA LOGICO CREADO.	FUNCION SELECCIONADA	

Figura 4.17. Narrativa del módulo Graficación Individual.

MÓDULO: CREACION DE DIAGRAMA LOGICO.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCION	ENTRADA	SALIDA
IDENTIFICA Y CREA VARIABLES MINIMAS	IDENTIFICA LAS VARIABLES QUE DEBEN APARECER EN EL DIAGRAMA Y LAS CREA.	APUNTADES A ESTRUCTURA TEAM_MINI_MAXI	
CREACION Y UNION DE COMPUERTAS	CREA LAS COMPUERTAS Y REALIZA LOS CALCULOS NECESARIOS PARA SER UNIDAS.	APUNTADES A ESTRUCTURA TEAM_MINI_MAXI	

Figura 4.18. Narrativa del módulo Creación de Diagrama Lógico.

IV.2. DISEÑO ARQUITECTÓNICO

MÓDULO: CREACION Y UNION DE COMPUERTAS.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCION	ENTRADA	SALIDA
IDENTIFICA Y CREA LINEAS	IDENTIFICA LAS LINEAS QUE CONECTAN A LAS COMPUERTAS Y LAS CREA.	APUNTAADORES A ESTRUCTURA TEAN_MINI_MAXI	APUNTAADORES A ESTRUCTURA CIRCUITO
IDENTIFICA Y CREA COMPUERTAS.	IDENTIFICA LAS COMPUERTAS QUE SE REQUIEREN EN EL DIAGRAMA LOGICO Y LAS CREA.	APUNTAADORES A ESTRUCTURA TEAN_MINI_MAXI	APUNTAADORES A ESTRUCTURA CIRCUITO
ARCHIVO GRAFICO	CREA Y MANEJA UN ARCHIVO EN EL CUAL SE ALMACENA LA INFORMACION DEL DIAGRAMA LOGICO.	APUNTAADORES A ESTRUCTURA CIRCUITO	APUNTAADOR DE ARCHIVO DIAGRAMA LOGICO

Figura 4.19. Narrativa del módulo Creación y Unión de Compuertas.

MÓDULO: ARCHIVO GRAFICO.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCION	ENTRADA	SALIDA
CREACION	CREA EL ARCHIVO DEL DIAGRAMA LOGICO.	DATOS GRAFICOS	APUNTAADOR DE ARCHIVO DIAGRAMA LOGICO
MANEJO	MANIPULACION DEL ARCHIVO DEL DIAGRAMA LOGICO.		

Figura 4.20. Narrativa del módulo Archivo Gráfico.

MÓDULO: DESPLIEGUE DE DIAGRAMA LOGICO.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCION	ENTRADA	SALIDA
INICIALIZA MONITOR GRAFICO	VERIFICA TIPO DE MONITOR Y LO INICIALIZA EN FORMA GRAFICA.		TIPO_MONITOR
CONTROL DE MOVIMIENTO	PERMITE AL USUARIO CONTROLAR EL DESPLIEGUE DEL DIAGRAMA.	APUNTAADOR DE ARCHIVO DIAGRAMA LOGICO	

Figura 4.21. Narrativa del módulo Despliegue de Diagrama Lógico

MÓDULO: GRAFICACION GLOBAL.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCION	ENTRADA	SALIDA
CREACION DE DIAGRAMA LOGICO	CREA EL DIAGRAMA LOGICO DE LA FUNCION.	RENTADORES A ESTRUCTURA YEAR_MINI_MAXI	RENTADOR DE ARCH. DE DIA-GRAMA LOGICO
DESPLIEGUE DE DIAGRAMA LOG.	DESPLIEGA EL DIAGRAMA LOGICO CREADO.		

Figura 4.22. Narrativa del módulo Graficación Global.

MÓDULO: IMPRESION.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCION	ENTRADA	SALIDA
IMPRESION INDIVIDUAL	REALIZA LA IMPRESION DEL DIAGRAMA LOGICO DE UNA DE LAS FUNCIONES MINITIZADAS CON SU CORRESPONDIENTE RESULTADO.	NO. DE FUNCION PAGINA INICIAL PAGINA FINAL	
IMPRESION GLOBAL	REALIZA LA IMPRESION DEL DIAGRAMA LOGICO DE TODAS LAS FUNCIONES MINITIZADAS CON SU CORRESPONDIENTE RESULTADO.	PAGINA INICIAL PAGINA FINAL	

Figura 4.23. Narrativa del módulo Impresión.

MÓDULO: IMPRESION INDIVIDUAL.			
PROCESOS		PARAMETROS	
NOMBRE	DESCRIPCION	ENTRADA	SALIDA
IMPRESION DE DIAGRAMA LOGICO	REALIZA EL PROCESO DE IMPRESION.	RENTADOR DE ARCHIVO DIA-GRAMA LOGICO	
SELECCION DE FUNCION	PERMITE AL USUARIO SELECCIONAR UNA DE LAS FUNCIONES A SER IMPRESA.	NO. DE FUNCION	

Figura 4.24. Narrativa del módulo Impresión Individual.

MÓDULO: IMPRESIÓN DE DIAGRAMA LÓGICO.			
PROCESOS		PARÁMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
ALMACENAMIENTO TEMPORAL DE IMPRESIÓN	INTRODUCE AL ALMACENAMIENTO TEMPORAL LOS DATOS CORRESPONDIENTES AL DIAGRAMA LÓGICO PARA SER TRANSMITIDOS.	NO. DE PÁGINAS A IMPRIMIR	
VERIFICACIÓN DE IMPRESIÓN	VERIFICA QUE LA IMPRESORA CUMPLA CON TODAS LAS CONDICIONES PARA POSTERIORMENTE IMPRIMIR.	CÓDIGO DE OPERACIÓN DE IMPRESORA	
VACIADO DEL ALMACENAMIENTO	TRANSMISIÓN DEL ALMACENAMIENTO TEMPORAL A LA IMPRESORA PARA IMPRIMIR EL DIAGRAMA.		

Figura 4.25. Narrativa del módulo Impresión de Diagrama Lógico.

MÓDULO: IMPRESIÓN GLOBAL.			
PROCESOS		PARÁMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
IMPRESIÓN DE DIAGRAMA LÓGICO	REALIZA EL PROCESO DE IMPRESIÓN.	APUNTADEOR DE ARCHIVO DE DIAGRAMA LÓGICO	

Figura 4.26. Narrativa del módulo Impresión Global.

MÓDULO: OPCIONES.			
PROCESOS		PARÁMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
ELIMINAR	REALIZA LA ELIMINACIÓN PARCIAL O TOTAL DE LA FUNCIÓN DE ENTRADA.		APUNTADEOR DE ARCHIVO DE FUNCIONES
INSERTAR	REALIZA LA INSERCIÓN PARCIAL O TOTAL DE LA FUNCIÓN DE ENTRADA.		APUNTADEOR DE ARCHIVO DE FUNCIONES

Figura 4.27. Narrativa del módulo Opciones.

IV.2. DISEÑO ARQUITECTÓNICO

MÓDULO: ELIMINAR.			
PROCESOS		PARÁMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
FUNCION	ELIMINA UNA DE LAS FUNCIONES QUE HAN SIDO INTRODUCIDAS.	NO. DE FUNCION	APUNTAADOR DE ARCHIVO DE FUNCIONES
TERMINO	REALIZA LA ELIMINACION DE TERMINOS DE UNA DE LAS FUNCIONES QUE HA SIDO INTRODUCIDA.	TERMINO NO. DE FUNCION	APUNTAADOR DE ARCHIVO DE FUNCIONES

Figura 4.28. Narrativa del módulo Eliminar.

MÓDULO: INSERTAR.			
PROCESOS		PARÁMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
FUNCION	CAPTURA NUEVAS FUNCIONES AL SISTEMA.	NO. DE FUNCION	APUNTAADOR DE ARCHIVO DE FUNCIONES
TERMINO	CAPTURA NUEVOS TERMINOS A LAS FUNCIONES YA EXISTENTES.	TERMINO NO. DE FUNCION	APUNTAADOR DE ARCHIVO DE FUNCIONES

Figura 4.29. Narrativa del módulo Insertar.

MÓDULO: CONTROL DE VENTANAS.			
PROCESOS		PARÁMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
CREACION DE VENTANAS	CREA LAS VENTANAS QUE SERAN DESPLEGADAS.	NO. DE VENTANA	APUNTAADORES A ESTRUCTURA REHUVENTANA
DESPLIEGUE DE VENTANAS	GENERA EN PANTALLA LA VENTANA SELECCIONADA.		
ELIMINACION DE VENTANAS	ELIMINA LA VENTANA QUE HA SIDO DESPLEGADA.	NO. DE VENTANA	

Figura 4.30. Narrativa del módulo Control de Ventanas.



MÓDULO: DESPLIEGUE DE VENTANAS.			
PROCESOS		PARÁMETROS	
NOMBRE	DESCRIPCIÓN	ENTRADA	SALIDA
ESCRITURA	PERMITE ESCRIBIR EN LA VENTANA.	TEXTO	
LECTURA	ANALIZA CÓDIGO DE TECLA PULSADA.		
BORRADO	BORRA LA VENTANA DESPLEGADA.		

Figura 4.31. Narrativa del módulo Despliegue de Ventana.

#### IV.3. DISEÑO DETALLADO.

En el diseño detallado se describe el flujo de información, su representación y detalles algorítmicos dentro de cada módulo. Se emplea un pseudocódigo estructurado para especificar este punto.

La figura 4.32 es el pseudocódigo correspondiente al módulo de Determinación Primos Implicantes, éste representa la forma en que se desarrolla cada uno de los módulos descritos en la sección anterior.

Se utiliza éste módulo para especificar el diseño, porque es la parte principal del sistema, sin éste proceso no es posible realizar otra aplicación a excepción de opciones.

##### Función Implicantes

Propósito: Seleccionar los primos implicantes de las funciones de Boole

Ejemplo de llamada: implicantes()

Entradas: Ninguna

INICIALIZAR variables y archivos  
 ABRIR archivos de trabajo  
 ASIGNAR dirección de funciones en el archivo

## REPETIR

INICIALIZAR variables  
 ABRIR archivos de trabajo  
 REPETIR

VERIFICAR adyacencia entre grupos de términos

SI los grupos son adyacentes ENTONCES

INCREMENTAR siguiente grupo

EXTRAER el primer grupo a combinar

INICIALIZAR apuntadores del primer grupo

ASIGNAR dirección del segundo grupo

CALCULAR posición del siguiente grupo

SI posición es igual a -1 ENTONCES

CALCULAR posición final del grupo dos

FIN SI

MIENTRAS existan términos en el grupo uno  
 y no se aborte el proceso REPETIR

MIENTRAS la dirección del grupo dos sea menor al  
 siguiente grupo REPETIR

EXTRAER término del segundo grupo

SI el término esta marcado ENTONCES

inicializar bandera con verdadero

SI NO

INICIALIZAR bandera con falso

FIN SI

VERIFICAR etiquetas de funciones

SI son etiquetas iguales ENTONCES

SI bandera de segunda lista es verdadera ENTONCES

SI potencia del grupo uno es igual a potencia  
 del grupo2 ENTONCES

LLAMAR módulo de desición para insertar término

FIN SI

SI NO

LLAMAR módulo de desición para insertar término

FIN SI

FIN SI

SI el término del grupo uno no esta marcado y su  
 bandera de marca es falsa ENTONCES

MARCAR el término en archivo

FIN SI

ELIMINAR el término del grupo dos

FIN MIENTRAS todos los términos del grupo dos  
 no esten procesados

ASIGNAR apuntador del primer grupo

SI el término del primer grupo no esta marcado ENTONCES

LLAMAR módulo que crea la siguiente lista

FIN SI

```

ELIMINAR el término del grupo uno
ASIGNAR dirección del proximo término del grupo uno
ASIGNAR dirección del grupo dos
LLAMAR al módulo que verifica si aborta proceso
SI abortar el proceso es verdadero ENTONCES
    LLAMAR al módulo que aborta el proceso
FIN SI
FIN MIENTRAS existan términos en el grupo uno
INICIALIZAR grupo uno con el grupo dos
SI NO
    SI existen términos en el grupo uno y si el
    grupo uno es menor que el total de grupos ENTONCES
        LLAMAR módulo que genera el grupo uno
        INICIALIZAR apuntadores
    FIN SI
FIN SI
FIN SI
SI no se abortó el proceso ENTONCES
    LLAMAR módulo que verifica si aborta proceso
    SI abortar el proceso es verdadero ENTONCES
        LLAMAR al módulo que aborta el proceso
FIN MIENTRAS los grupos sean adyacentes
MIENTRAS existan términos en el grupo uno repetir
    SI el término no tiene marca ENTONCES
        LLAMAR módulo que crea lista de primos
    ELIMINAR el término
FIN SI
FIN MIENTRAS no estén procesados todos los términos
SI existe lista de primos ENTONCES
    INICIALIZAR variables
    CERRAR archivo
    CREAR nuevo archivo
    RENOMBRAR archivo nuevo
FIN SI
FIN MIENTRAS existan lista de primos
SI no se abortó el proceso ENTONCES
    SI la bandera de un solo archivo es verdadera ENTONCES
        LLAMAR modulo que crea archivo de primos
    SI NO
        LLAMAR proceso que crea archivo auxiliar de primos
    FIN SI
FIN SI
ELIMINAR archivos de trabajo
TERMINAR el programa

```

Figura 4.32. Seudocódigo del módulo Determinación Primos Implicantes

## CAPITULO V

### DESARROLLO DEL SISTEMA

En esta fase se selecciona el lenguaje de programación apropiado, que cubra con los requerimientos especificados por el sistema. Una vez seleccionado se inicia el proceso de traducción del pseudocódigo planteado en el diseño detallado a instrucciones que son ejecutadas por la computadora.

Cada módulo generado debe estar documentado. Para lograr este objetivo se utilizan identificadores en variables y constantes que reflejen su funcionamiento. Se realizan comentarios que proporcionen alguna idea del proceso que realiza el módulo.

#### V.1. ELECCION DEL LENGUAJE.

Dado que las características disponibles en el lenguaje ejercen una fuerte influencia sobre la estructura arquitectónica y los detalles algorítmicos del sistema, es necesario, cubrir lo siguiente (FA187):

- Utilización de estructuras de datos
- Compilación por separado de módulos
- Tipos de datos definidos por el usuario
- Abstracción de datos
- Manejo de excepciones definidas por el usuario
- Facilite la creación de ventanas y menús encadenados

C se puede describir como un lenguaje de nivel medio debido a que contiene las características y capacidades de un lenguaje de alto y bajo nivel.

C tiene incluidas todas las características citadas anteriormente además de la velocidad y flexibilidad del lenguaje ensamblador así como las estructuras de datos y de control (if - else, for, do y while) de un lenguaje de alto nivel. También podemos citar entre otras las siguientes características :

- La transportabilidad
- Modularidad de los programas
- Permite crear estructuras de datos complejas
- Manipulación de bits, bytes y direcciones de memoria
- Funciones gráficas de biblioteca
- Incluye manejo de apuntadores y capacidades aritméticas de direcciones.

## V.2. CODIFICACION.

En esta etapa se traducen los pseudocódigos estructurados que fueron desarrollados en el diseño detallado. A continuación se describe el código fuente del módulo Determinación Primos Implicantes con las características que se emplearon para toda la codificación del sistema.

Algunas de las funciones que utiliza pertenecen al medio ambiente del lenguaje y otras fueron diseñadas.

```

/* Función Implicantes */
/* Propósito: Seleccionar los primos implicantes de las funciones
de Boole */
/* Ejemplo de llamada: implicantes() */
/* Entradas: Ninguna */
/* Salidas: 1) AbortaProceso
2) DirTermArch
3) NumCostoPrim
4) DirecPrimCosto
5) PrimImp
6) TermFunc */

char implicantes()
{
    int PosiUno, /* Posición del máximo número de unos */
        Grup1Uno, /* Posición del grupo 1 a combinar */
        Grup2Uno, /* Posición del grupo 2 a combinar */
        PosFinGrup2; /* Posición Final del grupo 2 */

    int AuxPosUnoTot, /* Auxiliar de PosiUno */
        LongCad; /* Guarda la longitud de una cadena */

    char bandera, /* Bandera a partir de la lista 2 de
de implicantes */
        existe, /* Bandera de combinación para fun--
ciones múltiples */
        Inicio, /* Bandera de inicio del grupo 1 */
        Adyacente; /* Bandera de adyacencia entre grupoi
y grupo 2 */

    char BandMarca, /* Bandera de indicación para marcar */
        alguna combinación de primos

```

```

NewArch,          /* Bandera de creación de un nuevo
                  archivo de primos implicantes */
AbortaProc,      /* Bandera de abortar el proceso */
UnoArch[3];      /* Arreglo para almacenar No. de unos */

FILE *ArchApunt, /* Apuntador al archivo de funciones */
*ApuntFunc,      /* Apuntador al archivo de términos */
*AuxApuntFunc;   /* Apuntador al archivo de listas */

long LongArch,   /* Longitud de algún archivo */
DirecGrup2;     /* Dirección del grupo2 en el archivo*/

PFUNCTION ProxTermino; /* Apuntador a la estructura de terminos */

/* Inicialización de variables y condiciones iniciales */

setmem(ApuntTab, sizeof(ApuntTab), -1);

AbortaProc = FALSE;
ApuntFunc = fopen("ArchFunc", "rb");
AuxApuntFunc = fopen("IniTab", "wb");
CopiaArch(ApuntFunc, AuxApuntFunc);
ApuntFunc = fopen("ArchTerm", "rb");
AuxApuntFunc = fopen("TermFunc", "wb");
CopiaTermFuncArch(ApuntFunc, AuxApuntFunc);
ListaSigDisc = FALSE;
libera();
AuxPosUnoTot = PosUnoTot;
numprim = 0;
MemoLibre = 0;
setmem(DirTermArch, sizeof(DirTermArch), -1);
setmem(DirecPrimCosto, sizeof(DirecPrimCosto), -1);
setmem(NumCostoPrim, sizeof(NumCostoPrim), 0);
setmem(MemoriaTerm, sizeof(MemoriaTerm), 0);
BandTermFunc = FALSE;
MemUsada = 0;
ArchApunt = fopen("PrimImp", "ab");
ApuntFunc = fopen("TermFunc", "r+b");
AsignaArreglo(DirTermArch, DirecFunc, Funciones);

/* Cuerpo principal */

do
{
crea = FALSE;
Inicio = TRUE;
Grup1Uno = 0;
AuxApuntFunc = fopen("IniTab", "r+b");
LongArch = ApuntTab[PosUnoTot + 1];

```

```

do
(
  /* Verificación de adyacencia entre grupos a combinar */
  Adyacente = AdyGrup(&Inicio, &Grup1Uno, bandera, ArchApunt,
  ApuntFunc);
  if (Adyacente)
  {
    Grup2Uno = Grup1Uno + 1;
    GenGrupLista(Grup1Uno, AuxApuntFunc, LongArch);
    elem_eli = pinicio;
    pinicio = pultimo = NULL;
    DirecGrup2 = ApuntTab[Grup2Uno];
    PosFinGrup2 = AsigPosUnoAbajo(Grup2Uno + 1, ApuntTab, PosUnoTot);
    if (PosFinGrup2 == -1)
      PosFinGrup2 = PosUnoTot + 1;
    while (elem_eli != NULL && !AbortaProc)
    {
      setmem(UnoArch, sizeof(UnoArch), '\0');
      fsetpos(AuxApuntFunc, &DirecGrup2);
      while (DirecGrup2 < ApuntTab[PosFinGrup2])
      {
        pininsert = (PFUNCION) malloc(sizeof(FUNCION));

        LongCad = lect_tam_arch(AuxApuntFunc);
        lect_arch(UnoArch, LongCad, AuxApuntFunc);
        LectApuntGrup(AuxApuntFunc, pininsert, LongCad, UnoArch);
        if (*pininsert->marca == '1')
          BandMarca = TRUE;
        else
          BandMarca = FALSE;
        pininsert->pder = NULL;
        pininsert->pizq = NULL;

        /* Verificar etiquetas de funciones para combinarlas */
        existe = (bin_dec(elem_eli->num_func) &&
        bin_dec(pininsert->num_func) > 0);
        if (existe)
          if (bandera)
          {
            if (strcmp(pininsert->TermElim, elem_eli->TermElim) == 0)
              decide_insert(bandera, Grup1Uno, ArchApunt, ApuntFunc);
          }
          else
            decide_insert(bandera, Grup1Uno, ArchApunt, ApuntFunc);

        /* Marcar el primo implicante en el archivo */
        fgetpos(AuxApuntFunc, &DirecGrup2);

```

```

    if (*pinsert->marca == '1' && !BandMarca)
        MarcaArchivo(AuxApuntFunc, DirecGrup2 - 1, '1');
    LibListPrimos(pinsert);
} /* while DirecGrup2 */
ProxTermino = elem_eli->pder;
if (*(elem_eli->marca) == '0')

    /* Creación de la próxima lista de primos implicantes */

    crea_list_tab(elem_eli->val_dec, elem_eli->TermElim,
        elem_eli->num_func, TABPRIM, bandera, Grup1Uno,
        ArchApunt, ApuntFunc); LibListPrimos(elem_eli);
elem_eli = ProxTermino;
DirecGrup2 = ApuntTab[Grup2Uno];

/* Verificar si hay interrupción para abortar el proceso */

AbortaProc = ProcesoAbortado();
if (AbortaProc)
    IniciaAbortoImp(&Adyacente, &crea, AuxPosUnoTot);
} /* while elem_eli */
Grup1Uno = Grup2Uno;
Inicio = FALSE;
} /* if adyacente */
else
{
    if (elem_eli == NULL && Grup1Uno <= PosUnoTot)
    {
        GenGrupLista(Grup1Uno, AuxApuntFunc, LongArch);
        elem_eli = pinicio;
        pinicio = pultimo = NULL;
    }
} /* else de adyacente */
if (!AbortaProc)
{
    AbortaProc = ProcesoAbortado();
    if (AbortaProc)
        IniciaAbortoImp(&Adyacente, &crea, AuxPosUnoTot);
}
}while (Adyacente);
while (elem_eli != NULL)
{
    if (*(elem_eli->marca) == '0')
        crea_list_tab(elem_eli->val_dec, elem_eli->TermElim,
            elem_eli->num_func, TABPRIM, bandera,
            Grup1Uno, ArchApunt, ApuntFunc);
    ProxTermino = elem_eli->pder;
    LibListPrimos(elem_eli);
    elem_eli = ProxTermino;
}
}

```



```

if (crea)
{
    /* Creación del archivo con la nueva lista de primos */

    bandera = TRUE;
    if (!ListaSigDisc)
        setmem(AuxApuntTab, sizeof(AuxApuntTab), -1);
    PosUnoTot = ArchTabla(TRUE, "");
    ListaSigDisc = FALSE;
    AsignaArreglo(ApuntTab, AuxApuntTab, variables + 2);
    fclose(AuxApuntFunc);
    remove("IniTab");
    rename("ProxTab", "IniTab");
} /* if del crea */
}while (crea); /* fin del do # 1 */
fclose(ArchApunt); fclose(ApuntFunc);

/* Verifica si no hay interrupciones */

if (!AbortaProc)
{
    if (!BandTermFunc)
        NuevoTermFunc();
    else
    {
        NewArch = FALSE;
        for (PosiUno = 0; PosiUno < Funciones; PosiUno++)
            if (MemoriaTerm[PosiUno] != 0)
            {
                OrdenaFuncMin(1, 1, MemoriaTerm[PosiUno], MemoriaTerm[PosiUno],
                    PosiUno, FALSE);

                MemoriaTerm[PosiUno] = 0;
                NewArch = TRUE;
            } /* if MemoriaTerm */
        if (NewArch)
        {
            ApuntFunc = fopen("TermFunc", "r+b");
            ApuntFunc = IntTermArchList(ApuntFunc, "TermFunc", "LlenaTab",
                TERMGLO);
            fclose(ApuntFunc);
        }
    } /* else */
    PosUnoTot = AuxPosUnoTot;
} /* if de !AbortaProc */
remove("IniTab");
return AbortaProc;
}

```

**PRUEBAS Y MANUAL DE USUARIO****VI.1 PRUEBAS.**

La prueba es un proceso de ejecución de un programa con la intención de descubrir la mayor cantidad de defectos. Para realizarlo, se usan dos técnicas de diseño de pruebas:

1. Prueba de la caja blanca.
2. Prueba de la caja negra.

Las pruebas de la caja blanca se centran en la estructura de control de programa. Se derivan casos de prueba que aseguren que se han ejecutado por lo menos una vez todas las sentencias del programa y que se ejercitan todas las condiciones lógicas.

Las pruebas de la caja negra son diseñadas para validar los requerimientos funcionales sin incluir el funcionamiento interno del programa. Las técnicas a utilizar se centran en el dominio de información de un programa, de forma que proporcione una cobertura completa de prueba. La partición equivalente divide el dominio de entrada en clases de datos que tienden a ejercitar determinadas funciones del programa.

De acuerdo a la figura 4.1 el "Sistema de Minimización" es dividido en tres módulos que interactúan para cumplir cualquier función del sistema. Cada uno de ellos son:

- 1) Control de menús.
- 2) Control de funciones.
- 3) Control de ventanas.

Cabe aclarar que la técnica de la caja blanca se verifica directamente en cada uno de los módulos utilizando como herramienta el debugger que el lenguaje C tiene incorporado en su medio ambiente. Se aplicó en todos los módulos del sistema corrigiendo cada uno de los errores detectados y optimizando el tiempo en el desarrollo de las pruebas manualmente.

En los módulos de control de menús y control de ventanas se verificó la interacción de procesos en la creación de menús, ventanas, pantallas de captura, despliegue de menús, ventanas y mensajes de aviso y error.

La técnica de la caja negra se utilizó cuando se integraron todos los módulos y formaron una sola unidad. En esta unidad se verificaron cada uno de los resultados esperados. A continuación se mencionan algunas de las pruebas que causaron algún efecto en el diseño del sistema.

**Prueba 1.**

Realizar la captura de funciones que tienen más de 150 términos.

**Prueba 2.**

Realizar la minimización de funciones que tienen más de 150 términos.

**Prueba 3.**

Graficación de funciones minimizadas.

**Prueba 4.**

Impresión de funciones minimizadas.

La prueba 1 sirvió para detectar que había insuficiente memoria principal para almacenar todos los datos. Esto se debió a que todo el tiempo los datos permanecían en memoria. El detectar este error dió inicio a una nueva evaluación de requerimientos que permitiera manejar más información utilizando archivos.

Una vez que el problema fue resuelto se inició la prueba 2. De acuerdo a los nuevos requerimientos, el proceso de minimización tuvo una serie de ajustes para poder interactuar entre la memoria principal y secundaria. En ésta prueba se observó que el tiempo de procesamiento aumentó en determinados procesos. En el apéndice A se anexan algunos tiempos promedio para obtener la función minimizada.

La importancia de la prueba 3 radica en visualizar el diagrama lógico equivalente de las funciones introducidas al sistema. Para tal efecto se desarrollaron continuas pruebas y algoritmos que conforme mostraban mayores características se obtenía notables cambios en cada uno de los algoritmos generados.

La prueba 4 fué similar a la anterior, salvo que ahora se enfocó al dispositivo de impresión. Algunos de los algoritmos utilizados en graficación fueron compatibles para esta prueba.

**VI.2. MANUAL DE USUARIO.****Introducción.**

Esta sección proporciona los conocimientos necesarios para la comunicación entre el usuario y el "Sistema de Minimización".

Este sistema está diseñado para todas las áreas que realicen diseño digital. Realiza la minimización de una o varias funciones de Boole accedidas por disco o por teclado.

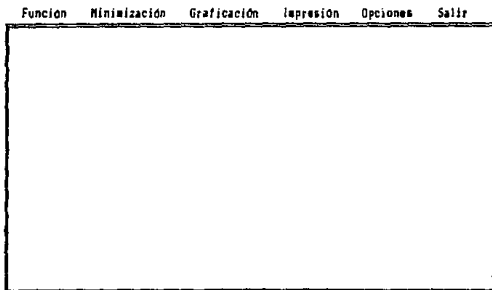
Tiene la ventaja de poder visualizar el resultado en forma canónica y gráfica, existiendo la opción de ser impresa.

La terminología empleada en el sistema es la usual en el área de electrónica digital y el usuario sólo debe tener los conocimientos básicos para usarlo.

#### Funcionamiento del Sistema.

Para entrar al sistema se teclea SM y < ENTER >. Se despliega en la pantalla la presentación con el título del paquete. El siguiente paso es presionar cualquier tecla para mostrar la figura 6.1. Las opciones ofrecidas por el sistema son :

- a) Función
- b) Minimización
- c) Graficación
- d) Impresión
- e) Opciones
- f) Salir



Fl-Ayuda Re Pág-Pant Arriba Av Pág-Pant Abajo ESC-Menu Ant

Figura 6.1 Menú principal del Sistema de Minimización.

Antes de describir el funcionamiento de cada opción es importante mencionar el modo de ayuda con que cuenta el sistema.

El Sistema de Minimización cuenta en todo momento con un sistema de ayuda en línea, que está ubicada en la parte inferior de la pantalla, para auxiliar al usuario en la función de teclas importantes.

F1 es una tecla de ayuda para el Sistema. Al ser presionada se presentará una ventana con la descripción de lo que realiza la opción en la que se encuentra ubicado el usuario. Por ejemplo, Si el usuario está posicionado en la opción Minimización, y pulsa la tecla F1, el sistema desplegará el mensaje correspondiente como se observa en la figura 6.2.

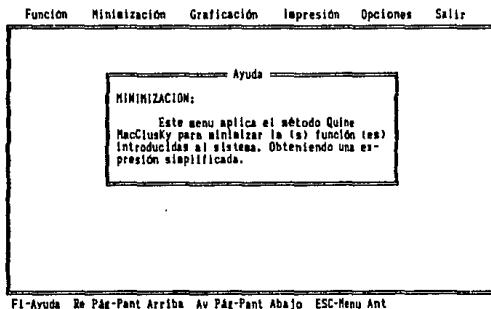


Figura 6.2 Texto de Ayuda de Minimización.

Existe un controlador de errores que despliega en una ventana textos que describen mensajes de error o de aviso, de acuerdo a la condición que suspenda la ejecución de algún proceso momentáneamente.

Todas las opciones con las que cuenta el Sistema pueden ser seleccionadas con sólo posicionarse en ella a través de las teclas ↑↓←→ y presionar < RETURN > o pulsando la primer letra del nombre de cada opción. Para salir de cualquier opción actual se presiona la tecla < ESC >. El modo de funcionamiento de cada una de las opciones es el siguiente :

a) **Función.** Esta opción es un menú que nos ofrece tres opciones :

Nueva  
Carga  
Salvar

a.1) **Nueva.** Esta opción es un submenú que permite la entrada de datos de una o más funciones en suma de productos o producto de sumas.

a.1.1) **Suma de productos.** Introduce los datos de las funciones de Boole en forma de suma de productos.

Al ser seleccionada ésta opción, se despliega en la pantalla la figura 6.3, donde se puede observar que es requerido teclear el número de funciones a minimizar, éstas funciones pueden variar de 1 a 15. Una vez accedado el número de funciones, el sistema pedirá el número de variables con que cuentan las funciones, éstas variables podrán variar desde 1 hasta 15, el sistema representa éstas variables con los caracteres < A > hasta la < O >. Después de introducir el número de variables, el sistema despliega en la parte derecha de la ventana en uso, el número de función que se encuentra en uso y el rango de entrada. El rango de

Función	Minimización	Graficación	Impresión	Opciones	Salir
Nueva					
Carga					
Salvar					
	Suma de Productos (Mint)				
	Producto de Sumas (Maxt)				
Función de Entrada					
No. de funciones : 1		Función : 1			
No. de variables : 6		Rango de entrada : 0 - 63			
Mintermino :					
F(A,B,C,D,E,F) = $\Sigma m(,)$					

Fl-Ayuda Re Pág-Pant Arriba Av Pág-Pant Abajo ESC-Menu Ant

Figura 6.3 Menú Nueva del Sistema de Minimización.

entrada representa el límite máximo de términos y términos no importa de la función.

A continuación el sistema pedirá los mintérminos y los despliega en la parte inferior. Al terminar de introducir los datos simplemente teclee < RETURN >. El sistema despliega un mensaje de aviso, figura 6.4, se pregunta si existen términos no importa, de ser afirmativo, se teclea < S > y los términos son capturados al igual que los mintérminos. Al concluir se presiona la tecla < RETURN >.

Para cualquier dato que haya sido teclado fuera del rango, previamente capturado o que no sea numérico, entre otros, el sistema despliega ventanas con mensajes de error.

Función	Minimización	Graficación	Impresión	Opciones	Salir
Nueva					
Carga					
Salvar					
	Suma de Productos (Mint)				
	Producto de Sumas (Maxi)				
Función de Entrada					
No. de funciones : 1		Funcion : 1			
No. de variables : 6		Rango de entrada : 0 - 63			
Términos no importa					
F(A, B, C, D, 47,45,51,5	Existen Términos No importa S/N : s				31,35,39,
F1-Ayuda Re Pág-Pant Arriba Av Pág-Pant Abajo ESC-Menu Ant					

Figura 6.4 Menú Nueva del sistema de Minimización.

- a.1.2) Producto de sumas. Captura los datos de la o las funciones de Boole en forma de producto de sumas.

La forma de introducir los datos es igual a la opción anterior, considerando que los datos son maxtérminos.

- a.2) Carga. Esta opción accesa los datos de la función o funciones a minimizar de un archivo con extensión bol.

Al ser seleccionada se despliega en la pantalla la figura 6.5. en la que se muestra el directorio de las funciones existentes.

Se debe posicionar en el nombre del archivo que se desee cargar por medio de las teclas  $\uparrow$   $\downarrow$   $\leftarrow$   $\rightarrow$  y teclear  $\langle$  RETURN  $\rangle$ .

a.3) **Salvar.** Salva en un archivo con extensión bol las funciones que fueron capturadas o modificadas recientemente.

Al ser seleccionada ésta opción el sistema pregunta el nombre que será asignado al archivo. Este nombre no debe ser mayor a ocho caracteres.

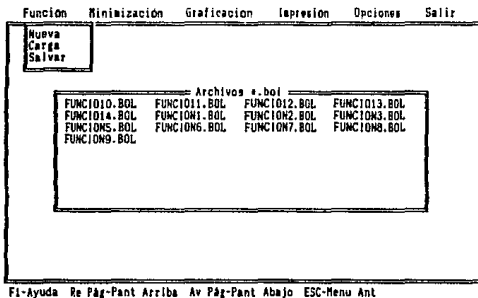


Figura 6.5 Menú Carga del Sistema de Minimización.

b) **Minimización.** Esta opción minimiza las funciones capturadas.

Al seleccionar ésta opción, el sistema despliega ventanas de aviso, en las cuales se expone los procesos que están ejecutando, como en la figura 6.6. Al concluir la minimización el resultado es desplegado en una ventana. Para salir al menú principal se presiona la tecla  $\langle$  ESC  $\rangle$ .

c) **Graficación.** Este es un menú que grafica el diagrama lógico de las funciones minimizadas. Si éstas no han sido



minimizadas manda un mensaje de aviso, para continuar se presiona cualquier tecla. Esta opción contiene las alternativas de graficación individual y global.

- c.1) Individual. Esta opción permite graficar en la pantalla el diagrama de sólo una de las funciones que hayan sido minimizadas.

Función	Minimización	Graficación	Impresión	Opciones	Salir
Funciones a Minimizar por Quine McClusky					PRUBIMP.BOL
PROCESO QUINE					63
No. de	Selección Primos Implicantes				
No. De	Pulse ESC para Abortar Proceso				
0					
1	1	1			
2	1	1			
16	1	1			
3	1	1			
5	1	X			
17	1	1			
18	1	1			
33	1	X			
48	1	X			

Fl-Ayuda Re Pág-Pant Arriba Av Pág-Pant Abajo ESC-Manu Ant

Figura 6.6. Menú Minimización del Sistema de Minimización.

Al ser seleccionada ésta opción el sistema preguntará el número de la función a graficar. Una vez capturado el sistema despliega en pantalla el diagrama lógico correspondiente.

Para el caso en el que la gráfica de la función no pueda ser observada en una pantalla, el usuario puede moverse horizontalmente y verticalmente a través de la teclas  $\uparrow$   $\downarrow$   $\leftarrow$   $\rightarrow$ .

- c.2) Global. Esta opción permite graficar el diagrama lógico de las funciones que han sido minimizadas.

Al igual que en la opción anterior la gráfica es manipulada a través de las teclas  $\uparrow$   $\downarrow$   $\leftarrow$   $\rightarrow$ .

- d) Impresión. Este menú permite imprimir el diagrama lógico de las funciones minimizadas.

El diagrama es impreso en una impresora de matriz de puntos configurada como IBM graphics. Si el diagrama a imprimir es mayor a una hoja, tanto horizontal como verticalmente, la impresión se realiza en la siguiente hoja y así sucesivamente.

Cuando el diagrama no pueda ser impreso o exista alguna interrupción, aparecen mensajes de aviso o de error correspondientemente.

El diagrama lógico es impreso en dos formas: Individual y Global, el proceso a seguir es igual que en graficación.

**e) Opciones.** Este es un menú que edita las funciones introducidas y permite la inserción y eliminación de funciones, mig términos, maxtérminos o términos no importa, según lo requerido por el usuario.

**e.1) Inserción.** Este submenú inserta funciones y términos.

**e.1.1) Término.** Esta opción permite insertar términos y términos no importa para la función o funciones existentes.

Al seleccionar ésta opción el sistema despliega la figura 6.7. en la que el sistema pide el valor del término a insertar, después de introducirlo, el sistema pide la función en la que será insertado, si existe más de una función. Para introducir el tipo de término que se desea insertar se tecleará un < 0 > para término y < 1 > para término no importa. Cuando exista algún problema en la inserción del término el sistema manda mensajes de aviso o de error.

**e.1.2) Función.** Esta opción permite insertar una función a las ya existentes.

Al seleccionar ésta opción el sistema despliega la figura 6.3. en la que se capturan los datos de la misma manera que en el inciso a.1.1.

**e.2) Eliminación.** Submenú que elimina funciones o términos que ya han sido introducidos.

**e.2.1) Término.** Esta opción permite eliminar términos y términos no importa para la función o las funciones existentes.

Al seleccionar ésta opción el sistema despliega la figura 6.8. en la que el sistema pide el número del

término a eliminar, después de introducirlo, el sistema pide la función en la que será eliminado.

Función	Minimización	Graficación	Impresión	Opciones	Salir
Funciones a Minimizar por Quine McClu					IMP.BOL
No. de Variables : 6		Rango d		Inserción	Eliminación
Numero Función		Mint :		Termino	Función
Decimal 0		0		:	X
Inserción de Terminos					
Termino : 10		Función : 1			
Termino : 1 ; No importa : 0 ; 1					
17	1	1			
18	1	1			
33	1		X		
48	1		X		

F1-Ayuda Re Pág-Pant Arriba Av Pág-Pant Abajo ESC-Menu Ant

Figura 6.7. Menú Inserción del Sistema de Minimización.

Función	Minimización	Graficación	Impresión	Opciones	Salir
Funciones a Minimizar por Quine McClu					IMP.BOL
No. de Variables : 6		Rango d		Inserción	Eliminación
Numero Función		Mint :		Termino	Función
Decimal 0		0		:	X
Eliminación de Terminos					
Termino : 10					
3	1	1			
5	1		X		
17	1	1			
18	1	1			
33	1		X		
48	1		X		

F1-Ayuda Re Pág-Pant Arriba Av Pág-Pant Abajo ESC-Menu Ant

Figura 6.8. Menú Eliminación del Sistema de Minimización.

Cuando exista algún problema en la eliminación del término el sistema proporcionará mensajes de aviso o de error.

e.1.2) **Función.** Esta opción permite eliminar una función de las ya existentes.

Al ser seleccionada el sistema pide el número de la función a ser eliminada.

f) **Salir.** Esta opción termina la sesión de trabajo. Recordándole al usuario si ya ha grabado su información en disco, con un mensaje de aviso, como el que se muestra en la figura 6.9.

Función	Minimización	Graficación	Impresión	Opciones	Salir
Funciones a Minimizar por Quine McClusky					PRUBINP.BOL
No. de Variables : 6		Rango de Entrada : 0 - 63			
Número Decimal	Función 0	Mint : 1 - No Imp : X 0			
0	1	1			
1	1	1			
2	1	1			
16	1	1			
----- Mensaje de Aviso -----					
Esta usted seguro de salir del sistema S/N : n					
48	1	X			

Fl-Ayuda Re Pág-Pant Arriba Av Pág-Pant Abajo ESC-Menu Ant

Figura 6.9. Menú Salir del Sistema de Minimización.

En la figura 6.10. se muestra la mínima expresión del ejemplo 4 utilizado en el inciso II.2, mapas de Karnaugh. El ejemplo es el siguiente :

$$F(A,B,C,D) = E_m(0,1,2,3,7,13,15,16,17,18,19,29,31,35,39,47,45, 51,55,59,63) + E_d(5,21,27,33,37,48,61,54)$$

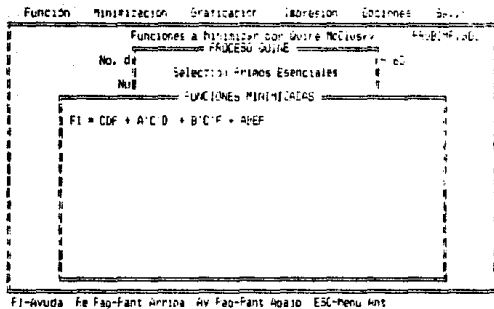


Figura 6.10. Despliegue de función minimizada.

En la figura 6.11. se muestra el diagrama lógico del mismo ejemplo.

Funciones Minimizadas : 1  
 No. de Variables : 6  
 Expresadas en Producto de Sumas  
 $F1 = ABEF + B'C'F + A'C'D + CDF$

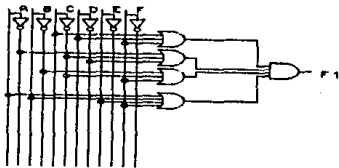


Figura 6.11. Diagrama lógico

Actualmente la computadora es una herramienta empleada para numerosas aplicaciones, de ahí es que cada vez desempeña un papel más importante en el desarrollo de la sociedad.

Aquí se empleo para desarrollar y usar el "Sistema de Minimización" para proporcionar una herramienta de apoyo al área digital, basado en el método Quine McCluskey seleccionado como la mejor alternativa por tener mayores ventajas sobre los métodos expuestos en el capítulo II.

Cabe mencionar que el método Quine McCluskey es un método sistemático con pasos bien definidos que realizan la búsqueda de términos que son seleccionados para encontrar una función simplificada. Al ser automatizado permite agilizar este proceso con resultados confiables, en el menor tiempo posible.

El haber realizado este sistema permite al usuario dedicarse a otro tipo de necesidades para no perder tiempo en obtener funciones de Boole mínimas y su correspondiente diagrama lógico.

Se cumplió el objetivo planteado cubriendo cada uno de los requerimientos especificados en la fase de análisis y en el momento que fueron detectados. Cada uno de los problemas que se encontraron fueron analizados obteniendo la mejor solución de acuerdo a las necesidades planteadas.

Entre las características que obtuvo el sistema se encuentran código reutilizable, facilidad de manejo, contar con textos de ayuda, es transportable y se puede ejecutar en computadoras personales IBM o compatibles de uno o dos discos flexibles, o en un disco duro con un monitor que tenga capacidad gráfica. Existiendo sus limitantes dentro de cada opción.

El sistema se puede ampliar en sus aplicaciones como es el caso de diseño de sistemas digitales utilizando circuitos secuenciales, los cuales serían procesados creando el diagrama lógico equivalente al circuito combinacional y al circuito secuencial.

Dado que no existen este tipo de herramientas al alcance del estudiante de Ingeniería, el sistema puede ser de gran utilidad por su facilidad de manejo y por su versatilidad para obtener un resultado confiable y seguro.

Con base en algunas entrevistas realizadas a profesores de diseño lógico se mostraron interesados en el sistema, ya que puede servir como herramienta de apoyo para los estudiantes de Ingeniería.

- (MOR83) Morris Mano, Arquitectura de computadoras, Prentice Hall, Colombia, 1983.
- (HIL85) J. Hill Frederick, R. Peterson Gerald, Teoría de conmutación y diseño lógico, Limusa, México, 1985.
- (MOR85) Morris Mano, Lógica digital y diseño de computadoras, Prentice Hall, Colombia, 1985.
- (IRW80) Irwin J. David, B. D. Carroll, An introduction to computer logic, Prentice Hall, N.J, 1982.
- (LEE82) C. Lee Samuel, Digital circuits and logic design, Prentice Hall, N.J. 1982.
- (PRE88) S. Pressman Roger, Ingeniería de software, McGraw-Hill, México, 1988.
- (FAI87) E. Fairley Richard, Ingeniería de software McGraw-Hill, México, 1987.
- (SAN85) Sanders Donal H., Informática: presente y futuro, McGraw-Hill, Estados Unidos, 1985.



## BIBLIOGRAFIA

---

- Morris Mano, Arquitectura de computadoras, Prentice Hall, Colombia, 1983.
- Hill Frederick J, R. Peterson Gerald, Teoria de conmutación y diseño lógico, Limusa, México, 1985.
- Morris Mano, Logica digital y diseño de computadoras, Prentice Hall, Colombia, 1985.
- Irwin J. David, B. D. Carrol, An introduction to computer logic, Prentice Hall, N.J, 1982.
- Lee Samuel C., Digital circuits and logic design, Prentice Hall, N.J. 1982.
- Pressman Roger S., Ingeniería de software, McGraw-Hill, México, 1988.
- Fairley Richard E., Ingeniería de software McGraw-Hill, México, 1987.
- Hear Donald, Baker M. Pauline, Gráficas por computadora, Prentice Hall-Hispanoamericana, 1988.
- Joltson Nelson, Advanced Graphics in C programming and techniques, McGraw-Hill, Berkeley California, 1987.
- Ezzell Ben, Graphics programing in Turbo C 2.0, Addison-Wesley, New York, 1989.
- Sanders Donal H., Informática: presente y futuro, McGraw-Hill, Estados Unidos, 1985.

FIGURAS.

1.1.	Diagrama General de un Sistema Digital	2
1.2.	Gráfica directa bipartita $F = (A+BC)(D+(BC)')(D+E)$	5
1.3.	Diagrama lógico de $F = (A+B)C' + D$	7
1.4.	Diagrama a bloques de un circuito combinacional	11
1.5.	Diagrama a bloques de un circuito secuencial	11
2.1.	Diagrama de Venn y mapa para dos variables	20
2.2.	Mapa de Karnaugh para 2,3,4,5 y 6 variables	22
2.3.	Tabla de verdad y su mapa correspondiente	23
2.4.	Mapa de Karnaugh de $F(A,B,C) = A'B+B'+A'BC$	23
2.5.	Mapa de Karnaugh de $F(A,B,C) = \Sigma m(0,2,4,7)$	24
2.6.	Mapa de Karnaugh de $F(A,B,C) = \Pi M(0,1,4,6)$	25
2.7.	Mapa con términos no importa	25
2.8.	Combinación de 1's para eliminar variables	26
2.9.	Agrupación de dos 1's en distintas formas	26
2.10.	Agrupación de cuatro 1's en distintas formas	27
2.11.	Agrupación de ocho 1's en distintas formas	27
2.12.	Implicantes de una función	28
2.13.	Primos implicantes esenciales	29
2.14.	Mapa para la función $F = B'C'D' + BCD + ACD' + A'B'C + A'BC'D$	30
2.15.	Mapa para la función $F = \Sigma m(2,3,4,7,9,11,12,13,14) + \Sigma d(1,10,15)$	30
2.16.	Mapa de Karnaugh para 5 variables	31
2.17.	Mapa de Karnaugh para 6 variables	31
2.18.	Mapa de Karnaugh para el ejemplo 3	32
2.19.	Mapa de Karnaugh para el ejemplo 4	32
2.20.	Mapa de Karnaugh para el ejemplo 5	33
3.1.	Funciones del "Sistema de Minimización"	51
3.2.	Diagrama formal del flujo de datos	52
4.1.	Carta estructurada del módulo principal	62
4.2.	Carta estructurada del módulo 2.1	63
4.3.	Carta estructurada del módulo 2.2	63
4.4.	Carta estructurada del módulo 2.3	64
4.5.	Carta estructurada del módulo 2.4	65

4.6.	Carta estructurada del módulo 2.5	65
4.7.	Narrativa del módulo Sistema de Minimización	66
4.8.	Narrativa del módulo Control de menús	66
4.9.	Narrativa del módulo Despliegue de menús	67
4.10.	Narrativa del módulo Control de funciones	67
4.11.	Narrativa del módulo Función de entrada	68
4.12.	Narrativa del módulo Nueva	68
4.13.	Narrativa del módulo Carga	69
4.14.	Narrativa del módulo Minimización	69
4.15.	Narrativa del módulo Proceso Quine	70
4.16.	Narrativa del módulo Graficación	70
4.17.	Narrativa del módulo Graficación Individual	71
4.18.	Narrativa del módulo Creación de Diagrama Lógico	71
4.19.	Narrativa del módulo Creación y Unión de compuertas	72
4.20.	Narrativa del módulo Archivo Grafico	72
4.21.	Narrativa del módulo Despliegue de diagrama lógico	72
4.22.	Narrativa del módulo Graficación global	73
4.23.	Narrativa del módulo Impresión	73
4.24.	Narrativa del módulo Impresión Individual	73
4.25.	Narrativa del módulo de Diagrama lógico	74
4.26.	Narrativa del módulo Impresión global	74
4.27.	Narrativa del módulo Opciones	74
4.28.	Narrativa del módulo Eliminar	75
4.29.	Narrativa del módulo Insertar	75
4.30.	Narrativa del módulo Control de ventanas	75
4.31.	Narrativa del módulo Despliegue de ventana	76
4.32.	Seudocódigo del módulo Determinación de primos implicantes	78
6.1.	Menú principal del Sistema de Minimización	87
6.2.	Texto de Ayuda de Minimización	88
6.3.	Menú Nueva del Sistema de Minimización	89
6.4.	Menú Nueva del Sistema de Minimización	90
6.5.	Menú Carga del Sistema de Minimización	91
6.6.	Menú Minimización del Sistema de Minimización	92
6.7.	Menú Inserción del Sistema de Minimización	94
6.8.	Menú Eliminación del Sistema de Minimización	94
6.9.	Menú Salir del Sistema de Minimización	95
6.10.	Despliegue de función minimizada	96
6.11.	Diagrama Lógico	96

## TABLAS

1.1.	Compuertas lógicas .....	3
1.2.	Tabla de verdad para n variables .....	4
1.3.	Tabla de verdad de la función $A'B'C + A'BC' + AB'C'$ .....	9
1.4.	Mintérminos y Maxtérminos para n = 3 variables .....	9
2.1.	Código Gray para 2 y 3 variables .....	21
2.2.	Agrupación de mintérminos de acuerdo a su número de unos .....	35
2.3.	Listados de primos implicantes .....	35
2.4.	Tabla de selección de primos implicantes ..	36
2.5.	Tabla de implicantes primos reducida .....	37
2.6.	Tabla de selección de primos implicantes .....	39
2.7.	Tabla de primos implicantes reducida .....	39
2.8.	Listados de primos implicantes .....	40
2.9.	Tabla de selección de primos implicantes esenciales .....	41
2.10.	Lista de primos implicantes .....	45
2.11.	Selección de primos implicantes .....	46
2.12.	Eliminación de columnas y renglones dominantes y dominados .....	47
2.13.	Selección de primos esenciales y primos esenciales secundarios .....	47
3.1.	Tabla de excepciones con mensajes de error ...	54
3.2.	Tabla de excepciones con mensajes de aviso ...	56