FACULTAD DE INGENIERIA ELECTRICA

ESCUELA POLITECNICA NACIONAL

"OPTIMIZACION DE CIRCUITOS LOGICOS CON AYUDA DEL COMPUTADOR DIGITAL EN LOGICA NAND DE HASTA TRES NIVELES".

Lucio V. Velasco P.

1976

CERTIFICO que el presente trabajo
ha sido realizado en su totalidad
por el señor LUCIO V. VELASCO P.

Sr. Ing. Jacinto Jijón
DIRECTOR DE TESIS

Octubre de 1.976

AGRADECIMIENTO

Expreso mi reconocimiento y gratitud a los señores Profesores Ingeniero Herbert Jacobson e Ingeniero Jacinto Jijón, por su valiosa ayuda como Directores de Tesis y al Cuerpo Docente de la Facultad de Ingeniería Eléctrica y Electrónica por haberme transmitido sus conocimientos a lo largo de toda mi carrera estudiantil.

INDICE GENERAL

INTRODUCCION 1

CRITERIO DE MINIMALIDAD 6

METODOS DE MINIMIZACION DE LA FUNCION BOOLEANA 8

- Método del Mapa de Karnaugh 8
- Método de Quine-McCluskey 10
- Método de Mott (Consensus) 19
- Método de Perlowski 29

ESTUDIO COMPARATIVO DE LOS PRINCIPALES METODOS 35

MINIMIZACION EN LOGICA NAND 38

- Método de Dietmeyer-Su 38
- Método de Muroga-Ibaraki 38
- Transformación de Lógica AND-OR a Lógica NAND 39

PROGRAMAS 42

- Programa correspondiente al Método de Quine-McCluskey 42
- Macrodiagrama de Flujo 43
- Guía del usuario 48
- Listados 50

- Programa correspondiente al Método de Mott 54
- Macrodiagrama de Flujo 55
- Cuía del usuario 60
- Listados 62

EJEMPLOS 68
- Ejemplo Nº 1 68
- Ejemplo Nº 2 75
- Ejemplo Nº 3 85

CONCLUSIONES 96

BIBLIOGRAFIA 98

INTRODUCCION

Las actuales presiones de la competencia en el mercado, obligan al diseñador a utilizar técnicas cada vez más eficaces de minimización. Dichas técnicas deben ser capaces de proporcionar resultados que se sometan a las especificaciones, restricciones y limitaciones impuestas por la tecnología moderna, la cual ha mostrado un interés cada vez creciente por el diseño con ayuda de un computador.

Si damos una mirada a la historia del diseño lógico, vemos que ésta va paralela a la historia del desarrollo de los circuitos integrados. Así en sus comienzos, el diseño lógico, estaba circunscrito a la utilización de compuertas lógicas individuales, construídas a partir de elementos concentrados tales como transistores, diodos, resistencias, etc., por lo cual una minimización tenía un caracter eminentemente práctico, ya que reducía considerablemente el costo de fabricación de un aparato lógico. Con la introducción de los circuitos integrados, esta técnica ha sido desplazada y actualmente el Ingeniero utiliza en sus diseños funciones dadas por módulos monolíticamente construídos y no ya circuitos de compuertas individuales.

Podemos apreciar la incidencia del desarrollo de los circuitos integrados, sobre el costo de fabricación de un sistema en las figuras 1a), 1b) y 1c).

La figura 1a) representa el número de circuitos integrados de un sistema lógico como función de la complejidad (número de e lementos de cada circuito integrado). En ella es evidente que la relación es lineal e inversamente proporcional.

El costo de cada circuito integrado como función de la comple jidad, está dado en la figura 1b). La curva (a) corresponde a circuitos "SSI" (SHORT SCALE INTEGRATION) y la curva (b) co - rresponde a circuitos "MSI" (MIDDLE SCALE INTEGRATION). Se ve claramente que en los circuitos integrados "MSI", se ha logra do extender la zona de costos bajos hasta niveles elevados de complejidad. Dicho en otras palabras, para un mismo costo, con los circuitos "MSI" se logra un número mucho mayor de compuertas por circuito integrado, que con la técnica "SSI".

En la figura 1c) tenemos las curvas correspondientes a cos- to de un sistema como función de la complejidad; la curva a) para circuitos "MSI" y la curva b) para "MSI". El mínimo pa- ra circuitos "MSI", es menor y está corrido hacia una mayor complejidad, lo que nos indica que el costo por compuerta se rá muchísimo menor con esta técnica que con la de los circui tos "SSI".

Veamos en forma práctica las diferencias arriba anotadas; un circuito tal como el CD001C, contiene cuatro compuertas lógi cas y su costo es de US$ 0.49 lo que nos da US$ 0.12 por com puerta; y un módulo tal como el CD4014C que contiene setenta y cinco compuertas y su costo es de US$ 1.51 con un costo por compuerta de US$ 0.02. Ha existido pues, una notable reducción del costo por compuerta al fabricar un circuito lógico mediante la técnica "MSI".

Existe una nueva técnica denominada "LSI"(LARGE SCALE INTEGRATION) que logra costos mínimos a elevadísimos niveles de complejidad. Aunque estos circuitos "LSI" son usados restringidamente todavía, nos dan una idea de la proyección que tiene en el futuro el Circuito Integrado.

Lo expuesto anteriormente nos lleva a una conclusión; en la actualidad una Minimización que tienda a encontrar expresiones mínimas con compuertas individuales es obsoleta, pero sigue el problema en vigencia por lo menos en el diseño de circuitos integrados "MSI" y mientras esto ocurra, la minimización de circuitos lógicos con ayuda de un computador, seguirá en vigencia.

Debe aclararse que aunque los métodos de minimización automática constituyen una efectiva ayuda, están siempre sujetos a serias limitaciones. Corroboran lo antedicho, el sinnúmero de trabajos realizados por estudiosos de la materia, sin lograrse aún, un algoritmo de caracter "universal" dentro de esta rama de la ciencia.

En el presente trabajo no se pretende agregar un algoritmo más a los ya existentes, sinó demostrar la factibilidad del uso del computador como herramienta de diseño lógico. Para cumplir este cometido, éste estudio estará dividido en dos fases: En la primera se intentará la Minimización de la función Booleana como tal y en la segunda la minimización en Lógica NAND.

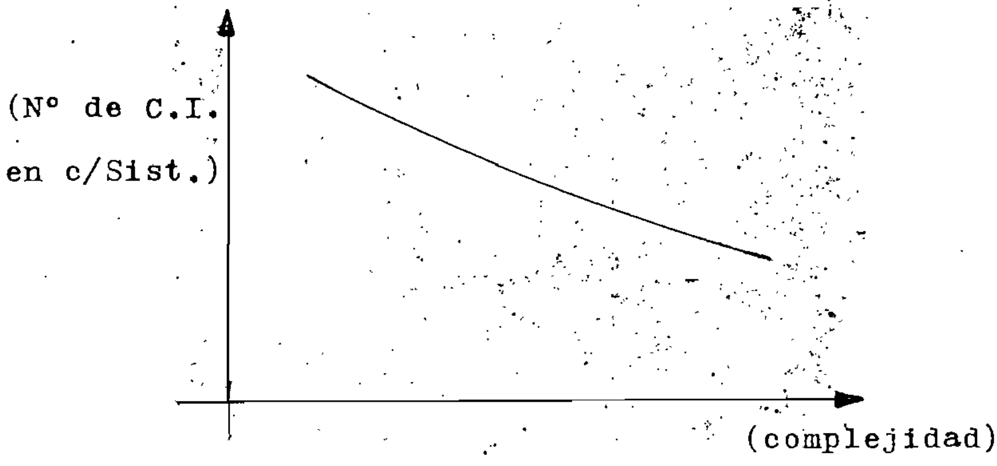


fig. 1a)

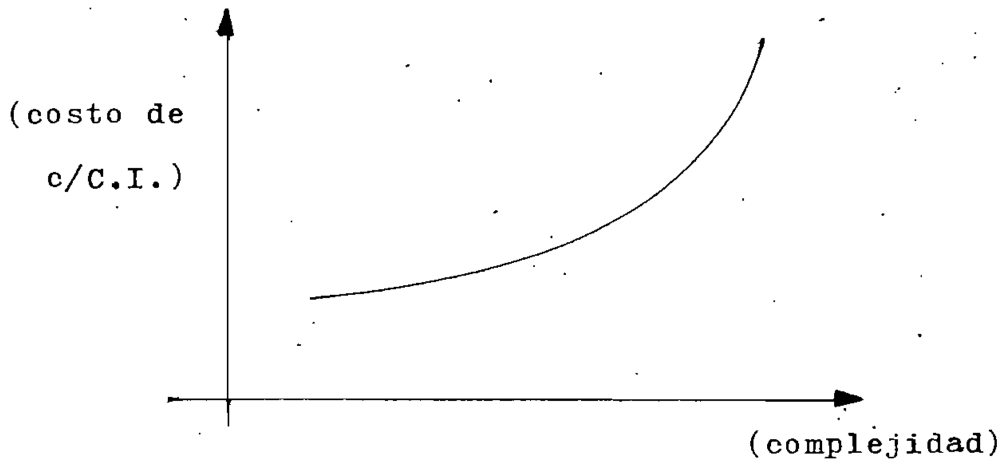


fig. 1b)

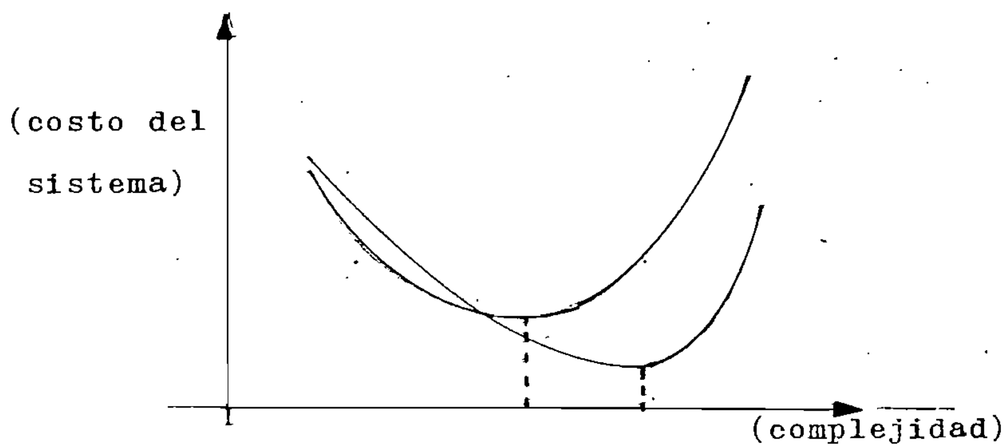


fig. 1c)

CRITERIO DE MINIMILIDAD

Existen varios criterios, los cuales tienden a determinar, cual de las expresiones equivalentes de una función lógica es mínima. Los criterios más usados son tres:

1.- La expresión mínima, es la expresión con el MENOR NUMERO DE LITERALES.

2.- La mínima expresión, es la expresión con el MENOR NUMERO DE TERMINOS, con tal que no exista otra expresión con el mismo número de terminos y menor número de literales.

3.- La expresión mínima es la expresión que requiere el MENOR NUMERO DE ELEMENTOS LOGICOS para su construcción. El número de elementos lógicos, está determinado por la suma del número términos con el número de literales contenidos en la función.

De estos tres criterios, el tercero, que contiene a los otros dos es el generalmente más usado. Se puede afirmar que todos los criterios provienen de un criterio más general y practico y que es el criterio del menor costo.

Veamos como inciden en la reducción de costos estos criterios.

Sea la función:

$$F = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2x_3 + x_1x_2\bar{x}_3 + x_1x_2x_3$$

que puede ser implementada por las siguientes expresiones:

1) $F = \bar{x}_1\bar{x}_2 + \bar{x}_1x_3 + x_2x_3 + x_1x_2$

2) $F = \bar{x}_1\bar{x}_2 + \bar{x}_1x_3 + x_1x_2$

3) $F = \bar{x}_1\bar{x}_2 + x_2x_3 + x_1x_2$

Tanto la expresión 2) como la 3) contiene el menor número de literales y el menor número de terminos; por lo tanto, de acuerdo al primero y segundo criterio estas dos expresiones son igualmente mínimas. Si queremos implementar la función utilizando elementos concentrados, necesitamos para la expresión 1), doce diodos y dos transistores; mientras que, para las expresiones 2 ó 3, solamente precisamos de nueve diodos y dos transistores. En consecuencia, nuevamente las expresiones 2) y 3) son igualmente mínimas, esta vez de acuerdo con el tercer criterio.

Es evidente que un número menor de elementos conlleva un menor costo, y por lo tanto, cualquiera de los criterios anotados tienden a reducir el costo. Sin embargo, cuando se trata de una construcción utilizando circuitos integrados, estos tres criterios no pueden ser extrictamente aplicados; limitaciones intrinsicas en número de entradas y compuertas en los modulos, dificultan la formulación de un criterio más general.

METODOS DE MINIMIZACION DE LA FUNCION BOOLEANA

Un estudio de algunos de los Métodos de Minimización de la Función Booleana como tal, será abordado en este capítulo. De éste ha de perfilarse la posibilidad de una programación de alguno ó algunos de los métodos.

METODO DEL MAPA DE KARNAUGH

Basicamente el Mapa de KARNAUGH, es una forma gráfica de representación de la misma información contenida en la tabla de verdad que define dicha función. Para una función de "r" variables, el mapa contiene 2^r cuadros, uno por cada combinación posible, de las "r" variables de entrada. A cada cuadro se designan valores particulares de dichas combinaciones. Bajo esta convención se coloca un "1" en cada cuadro que representa una combinación para la cual la función toma el valor uno (1); se coloca un cero (0) en cada cuadro que represente a una combinación para la cual la función toma el valor cero; y se coloca una "i" en aquellos cuadros correspondientes a condiciones de entrada "indefinidas" (Don't care).

EJEMPLO:

Sea $F(x_1,x_2,x_3)$, una función booleana representada por la tabla de verdad de la figura 2). El Mapa de Karnaugh para dicha función se ilustra en la figura 3)

Siguiendo ciertas reglas, es facil visualizar los agrupamien tos que nos lleven a una solución mínima.

TABLA DE VERDAD DE

UNA FUNCION BOOLEANA

x_1	x_2	x_3	F
0	0	0	1
0	0	1	1
0	1	0	i
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	i
1	1	1	0

fig. 2)

MAPA DE KARNAUGH DE

UNA FUNCION BOOLEANA

x_3 \ x_1x_2	00	01	11	10
0	1	i	i	0
1	1	1	0	0

fig. 3)

Como el número de combinaciones y por lo tanto el de cuadros crece exponencialmente con el número de variables (para 7 variables se requieren 128 cuadros), el método deja de ser práctico para seis o mas variables.

El método del mapa de Karnaugh no presenta características optimas para una programacion por dos razones: primero, el método es visual, y al programarlo pierde esta característica y segundo, demanda un consumo exagerado de tiempo y posiciones de memoria.

METODO DE QUINE - MCCLUSKEY

El método TABULAR o de QUINE - MCCLUSKEY, consiste en una técnica sistemática de enumeración basada en la relación

$$xy + x\overline{y} = x$$

donde "x" es cualquier expresión producto representando una ó más variables y "y" es una sola variable.

El proceso de minimización se lleva a cabo siguiendo el algoritmo detallado a continuación:

1.- Formar la expansión canónica en suma de productos de la función a minimizar.

2.- Examinar todos los productos y aplicar la reducción

$$xy + x\overline{y} = x$$

tantas veces como sea posible. Los nuevos términos así formados, tandrán una variable menos que los términos originales.

3.- Al nuevo conjunto de términos aplicar nuevamente el paso 2. Cuando ya no existan más reducciones posibles con los pasos 1 y 2, obtendremos el conjunto de implicantes primos asociados a la función a ser minimizada.

4.- De este conjunto de implicantes primos se obtendrá la mínima expresión de la función booleana.

A continuación se ilustra el método mediante un ejemplo.

Partimos del hecho de que disponemos de la tabla de verdad de la fig. 4). Con el fin de generalizar el método se ha escogido una función de cuatro variables, donde se ha incluído condiciones no implementadoras o indefinidas.

La expansión canónica de la función a ser minimizada es la siguiente:

$$F = \bar{x}_1\bar{x}_2x_3x_4 + \bar{x}_1x_2\bar{x}_3x_4 + \bar{x}_1x_2x_3x_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2x_3\bar{x}_4 + x_1\bar{x}_2x_3x_4 + x_1x_2\bar{x}_3\bar{x}_4 + x_1x_2\bar{x}_3x_4 + x_1x_2x_3x_4$$

Son condiciones indefinidas los productos,

$$\bar{x}_1\bar{x}_2x_3\bar{x}_4 \quad y \quad \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$$

TABLA DE VERDAD DE

UNA FUNCION BOOLEANA

x_1	x_2	x_3	x_4	F
0	0	0	0	i
0	0	0	1	0
0	0	1	0	i
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

fig. 4)

Para la ejecución del segundo paso, tabulamos los términos productos añadiendo a la lista, las condiciones indefinidad y aplicando al conjunto la relación

$$xy + x\overline{y} = x$$

tantas veces como fuese necesario. Comparemos pues, el primer término de la primera columna de la fig. 5), (0 0 0 0), con los demás términos de la misma columna; vemos que es posible aplicar la reducción solamente con el segundo y tercer término (0 0 1 0) y (1 0 0 0), obteniendo así los términos reducidos (0 0 - 0) y (- 0 0 0), a los que les colocamos en la co - lumna 2, de la misma figura. Continuamos la operación con todos los términos originales.

En el paso 3, a los nuevos terminos resultantes del paso anterior, aplicamos el principio de reducción hasta cuando no encontremos más posibilidades de simplificación obteniendo en consecuencia el conjunto de implicantes primos de la función tratada.

En la tabla de la fig.5) se muestran los resultados de los pasos 2. y 3.; en la primera columna se encuentra el conjunto original con las condiciones incluidas; en la segunda, los resultados de la primera reducción y en la tercera columna la reducción final. Se ha colocado una marca (V) en aquellos terminos que se han utilizado en alguna reducción y por lo tanto no constituyen implicantes primos. Se ahn señalado com (°) aquellos términos que no han tomado parte en una reducción, constituyendo en si, implicantes primos.

Con la tabla de implicantes primos obtenida en los pasos anteriores se procederá a determinar la minima expresión de la función original a ser minimizada.

PROCESO DE REDUCCION Y CALCULO

DE IMPLICANTES PRIMOS

x_1	x_2	x_3	x_4		PRIMERA REDUCCION x_1	x_2	x_3	x_4		SEGUNDA REDUCCION x_1	x_2	x_3	x_4	
0	0	0	0	V	0	0	–	0	V	–	0	–	0	°
0	0	1	0	V	–	0	0	0	V	–	0	1	–	°
1	0	0	0	V	0	0	1	–	V	–	–	1	1	°
0	0	1	1	V	–	0	1	0	V	–	1	–	1	°
0	1	0	1	V	1	0	–	0	V					
1	0	1	0	V	1	–	0	0	°					
1	1	0	0	V	0	–	1	1	V					
0	1	1	1	V	–	0	1	1	V					
1	0	1	1	V	0	1	–	1	V					
1	1	0	1	V	–	1	0	1	V					
1	1	1	1	V	1	0	1	–	V					
					1	1	0	–	°					
					–	1	1	1	V					
					1	–	1	1	V					
					1	1	–	1	V					

fig.5)

Para la determinación de la mínima expresión de la función, se seguirá el siguiente procedimiento:

a.- Formar la tabla de Selección.

b.- Determinar los implicantes primos esenciales.

c.- Determinar y eliminar aquellos implicantes que sean absolutamente eliminables.

d.- Seleccionar los Implicantes primos elegibles.

e.- Formar la Tabla Reducida de Selección.

f.- Seleccionar el grupo o grupos de implicantes primos que cubran la función en forma mínima.

La Tabla de Selección para el problema del ejemplo se ilustra en la fig.6).

TABLA DE SELECCION

IMPLICANTES PRIMOS	TERMINOS PRODUCTO								
	0011	0101	0111	1000	1010	1011	1100	1101	1111
1-00				v			v		
110-							v	v	
-0-0				v	v				
-01-	v				v	v			
--11	v		v			v			v
-1-1		v	v					v	v

fig. 6)

En la tabla de la fig. 6), cada columna corresponde a un término producto de la expresión original y cada fila corresponde a un implicante primo. Notese que no se han incluído los términos de condición "indefinida". Se ha colocado una marca "v" para cada implicante primo, en las columnas cuyos términos producto estan contenidos en ese implicante primo; así por ejemplo, el implicante primo (1-00) contiene a los términos producto (1000) y (1100).

De la misma tabla, podemos extraer el conjunto de implicantes primos esenciales, absolutamente eliminables y elegibles, tomando en consideración las siguientes observaciones:

- Un implicante primo que no contenga (-), contiene a un solo término producto; si contiene una (-) dos terminos producto estarán implicados por él. En general un implicante que contenga <u>s</u> (-), implicará a 2^s términos producto.
- Es posible tener uno ó más término producto, que estén contenidos en un solo implicante primo. Los implicantes primos que cumplan esta función se denominan implicantes primos esenciales. En el ejemplo, (-1-1) es esencial.
- Un implicante primo es absolutamente eliminable, si todas las columnas que han sido marcadas en la fila correspondiente a este implicante primo, están también marcadas en una o más filas correspondientes a los implicantes primos esenciales. En el ejemplo no existe ningún implicante primo que cumpla este requisito.

- Los implicantes primos, que no se encuentren en ninguna de las dos clases definidas anteriormente constituyen el conjunto de implicantes primos elegibles. Por lo tanto, son elegibles los implicantes: (1-00), (110-), (-0-0), (-01-) y (--11).

La tabla reducida de selección nos permite determinar que conjunto de implicantes representarán a la función su forma más simple. Esta tabla es la misma Tabla de Selección en la que constan solamente las filas y columnas correspondientes a los implicantes primos elegibles y a los términos productos implicados por éstos respectivamente. La fig. 7) ilustra a la tabla reducida de selección del ejemplo.

TABLA REDUCIDA DE SELECCION

IMPLICANTES PRIMOS	TERMINOS PRODUCTO 0011	1000	1010	1011	1100
1-00		v			v
110-					v
-0-0		v	v		
-01-	v		v	v	
--11	v			v	

fig. 7)

Para encontrar la mínima expresión con la ayuda de esta tabla, debemos seleccionar grupos de implicantes primos, de tal suerte que todos los términos producto que han formado parte en esta tabla, esten cubiertos por dichos grupos. Aquel grupo que unido al conjunto de implicantes primos esenciales cumpla con los criterios de minimalidad dados en el capítulo anterior constituirán la mínima expresión de la función. Procediendo de la manera indicada obtendremos en el ejemplo los siguientes grupos:

Grupo 1	(1-00), (-01-)
Grupo 2	(1-00), (-0-0), (--11)
Grupo 3	(110-), (-0-0), (-01-)
Grupo 4	(110-), (-0-0), (--11)

Obviamente existen otros grupos a más de los cuatro anteriores, por ejemplo, tomemos el grupo formado por los implicantes (110-), (-0-0), (-01-) y (--11). Si examinamos la fig. 7) vemos que la fila correspondiente al implicante primo (--11), está completamente cubierta por la fila correspondiente al implicante (-01-); por lo tanto, sería redundante considerar al implicante (--11) como miembro de este grupo.

El grupo 1 cumple con el tercer criterio de minimalidad y por lo tanto constituye junto con el implicante primo esencial (-1-1), la mínima expresión que representa la función en suma de productos, quedando esta como sigue:

$$F = x_2x_4 + x_1\bar{x}_3\bar{x}_4 + \bar{x}_2x_3$$

En muchos casos la selección de los grupos necesarios para completar la mínima representación de la función es obvia. Sin embargo, en algunas situaciones, se necesita de un algoritmo capaz de generar en forma sistemática el grupo de cobertura mínima. Taylos L. Booth presenta un algoritmo algébrico para la solución de este problema.

Este método como se puede apreciar en su estudio resulta bastante claro y sencillo, presentando buenas características para una posible programación.

METODO DE MOTT (CONSENSUS)

Por el mismo tiempo que Quine-MacCluskey desarrollaban el método tabular de minimización, Samson y Mills trabajaban sobre una nueva técnica denominada de Consensus, que Quine la define como CONSENSUS ITERADO. Más tarde Mott combinó el método de Consensus con el Tabular y consiguió una mínima expresión de la función booleana, al mismo tiempo que puso en evidencia la posibilidad de adaptar este método para la programación.

El Consensus Iterado consiste en dos operaciones básicas; una es la operación de consensus y otra es la relación de subsuma

Operación de Consensus.-

Si dos términos X y Y contienen una y solo una variable afirmada en el uno y negada en el otro, se puede escribir un nuevo término Z, de consensus, formado por el producto de todas las variables restantes, eliminando aquella que está afirmada en el uno y negada en el otro. Entonces se puede escribir que $X + Y = X + Y + Z$, donde el simbolo (=) significa equivalencia booleana.

Ejemplos:

$$x_1x_2\bar{x}_3 + x_1x_2x_3 = x_1x_2\bar{x}_3 + x_1x_2x_3 + x_1x_2$$

donde $x_1x_2 = Z$ (término de consensus)

$$x_1x_2 + \bar{x}_1x_3 = x_1x_2 + \bar{x}_1x_3 + x_2x_3$$

donde $x_2x_3 = Z$ (término de consensus)

No existe consensus cuando: 1) No hay un par de variables opuestas. Ejemplo:

$$x_1x_2 + x_2x_3$$

2) Cuando hay más de una variable opuesta. Ejemplo:

$$x_1\bar{x}_2x_3 + \bar{x}_1x_2x_3$$

<u>Relación de Subsuma.</u>-

Si cada variable de X está contenida en Y, y Y contiene variables adicionales, luego Y subsuma a X. Un término que subsuma a otro puede ser eliminado sin alterar la función, Ej:

$$F = x_1x_2x_3 + x_2x_3 + x_1x_2$$

$x_1x_2x_3$ subsuma a x_1x_2 y por lo tanto $x_1x_2x_3$ puede ser elimina -

de quedando,

$$F = x_2x_3 + x_1x_2$$

El consensus iterado, es un proceso por el cual una expresión dada puede ser transformada en un conjunto de implicantes primos. El procedimiento es llevado a cabo en dos pasos; en el primero se remueven los términos contenidos en otros (relación de subsuma), en el segundo, a los términos resultantes del primer paso se aplica la operación de consensus.

Para aclarar el método, apliquemoslo a un ejemplo, pero antes debemos hacer las siguientes anotaciones:

- Una variable afirmada es representada por <u>10</u>
- Una variable complementada por <u>01</u>
- Una variable ausente por <u>00</u>

La función que queremos minimizar es:

$$F = x_1\overline{x}_3 + x_1\overline{x}_2\overline{x}_3 + x_1\overline{x}_3\overline{x}_4 + \overline{x}_1\overline{x}_2\overline{x}_4 + \overline{x}_1x_2\overline{x}_3$$

Si hacemos uso del convenio anterior obtenemos la siguiente tabla . fig. 8), en la que cada fila representa un término producto.

Se va a tomar cada uno de los términos y se procederá a probar el criterio de Subsuma con todos los restantes. Si un término queda así eliminado, este ya no interviene en las siguientes pruebas.

TABLA DE TERMINOS PRODUCTO

CODIFICADOS

PRODUCTOS	VARIABLES x_1	x_2	x_3	x_4
$x_1\overline{x}_3$	10	00	01	00
$x_1\overline{x}_2\overline{x}_3$	10	01	01	00
$x_1\overline{x}_3\overline{x}_4$	10	00	01	01
$\overline{x}_1\overline{x}_2\overline{x}_4$	01	01	00	01
$\overline{x}_1x_2\overline{x}_3$	01	10	01	00

fig. 8)

Para esto escribimos cada pareja de términos de acuerdo a los codigos anteriores y procedemos a <u>sumar</u> verticalmente todos los símbolos individuales existentes, recordando que:

$0 + 0 = 0; \quad 0 + 1 = 1; \quad 1 + 1 = 1.$

Tomemos el primero y segundo término y apliquemos la opera - ción descrita a esta pareja:

10	00	01	00
10	01	01	00
10	01	01	00

con la consecución del segundo término reproducido en el re - sultado, lo que significa que está contenido en el primero y que por lo tanto podemos eliminarlo.

De igual manera eliminamos el tercer término $(x_1\bar{x}_3\bar{x}_4)$ y la función toma la forma de:

$$F = x_1\bar{x}_3 + x_1\bar{x}_2\bar{x}_4 + \bar{x}_1x_2\bar{x}_3$$

que en forma codificada está representada en la tabla de la fig. 9)

TABLA DE TERMINOS PRODUCTO

CODIFICADOS

PRODUCTOS	VARIABLES x_1	x_2	x_3	x_4
$x_1\bar{x}_3$	10	00	01	00
$x_1\bar{x}_2\bar{x}_4$	10	01	00	01
$\bar{x}_1x_2\bar{x}_3$	01	10	01	00

fig. 9)

al primero y segundo término de la tabla anterior, aplicamos la segunda parte del método y para ello sumamos en la forma ya descrita y obtenemos:

10	00	01	00
01	01	00	01
11	01	01	01

De la definición de operación de consensus podemos deducir que si existe solamente un término "11", existirá consensus y éste puede ser obtenido facilmente, reemplazando "11" por "00". En el ejemplo, $(\overline{x}_2\overline{x}_3\overline{x}_4)$ 00 01 01 01, será el término de consensus. A este término lo añadimos al final de la lista de la figura 9) y nuevamente aplicamos la primera parte del método, con el fin de eliminar, si existen, términos que subsumen a otros. La lista final, una vez que no existan términos de consensus y términos que subsumen a otros, es la lista de Implicantes Primos de la función tratada.

En nuestro problema del ejemplo la función queda reducida a:

$$F = x_1\overline{x}_3 + \overline{x}_3\overline{x}_4 + x_2\overline{x}_3$$

Si en el problema aparecen condiciones "indefinidas", éstas son incluidas y tratadas junto a los productos implementadores con el procedimiento anterior.

Una vez obtenido el conjunto de implicantes primos, se procede a determinar la mínima expresión de la función. Mott propone para la rsolución de este problema, la creación de una matriz denominada "Matriz de Cubrimiento, muy similar a la tabla de Selección del método tabular. Para formar esta matriz hay que expandir la función original a su forma conónica aplicando en ella la relación :

$$X = XY + X\overline{Y}$$

Realicemos la expansión del primer término de la función original del ejemplo;

$$x_1\bar{x}_3 = x_1x_2\bar{x}_3 + x_1\bar{x}_2\bar{x}_3 = x_1x_2\bar{x}_3x_4 + x_1x_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3x_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4$$

Continuando de la misma manera con los restantes términos, que no estén en su forma canónica, obtenemos la expansión de la función original como la ilustrada por la tabla de la fig. 10). En esta operación ya no intervienen los productos de condición indefinida.

TABLA DE TERMINOS PRODUCTO

DE LA FUNCION EXPANDIDA

	PRODUCTOS	VARIABLES x_1	x_2	x_3	x_4
1)	$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$	01	01	01	01
2)	$\bar{x}_1x_2\bar{x}_3\bar{x}_4$	01	10	01	01
3)	$\bar{x}_1x_2\bar{x}_3x_4$	01	10	01	01
4)	$x_1\bar{x}_2\bar{x}_3\bar{x}_4$	10	01	01	01
5)	$x_1\bar{x}_2\bar{x}_3x_4$	10	01	01	10
6)	$x_1x_2\bar{x}_3\bar{x}_4$	10	10	01	01
7)	$x_1x_2\bar{x}_3x_4$	10	10	01	10

fig. 10)

La Matriz de Cubrimiento, es una matriz M x N, donde M es el número de productos de la función expandida canónicamente y N es el número de Implicantes Primos de la función. Si m_{ij} es un elemento de dicha matriz y si X_j y Y_i son un término producto y un implicante primo respectivamente ($j = 1,\ldots,M$; $i = 1,\ldots,N$) entonces:

$$m_{ij} = \begin{cases} 1 \text{ si } Y_i \text{ cubre a } X_j \\ \\ 0 \text{ si } Y_i \text{ no cubre a } X_j \end{cases}$$

Para clarificar la forma de obtener cada elemento de la Matriz de Cubrimiento, se encontrarán los elementos m_{11} y m_{21}. La fig. 11) contiene al conjunto de implicantes primos de la función del ejemplo.

TABLA DE IMPLICANTES PRIMOS

	IMPLICANTES PRIMOS	x_1	x_2	x_3	x_4
1)	$x_1\bar{x}_3$	10	00	01	00
2)	$\bar{x}_3\bar{x}_4$	00	00	01	01
3)	$x_2\bar{x}_3$	00	10	01	00

fig. 11)

Investiguemos si el Implicante Primo 1)(fig. 11) cubre al término producto 1) (fig. 10), sumando dichos términos en la for ma acordada;

$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$	01	01	01	01
$x_1\bar{x}_3$	10	00	01	00
	11	01	01	01

El resultado nos indica que los elementos sumados no se sub - suman y por lo tanto, el implicante $x_1\bar{x}_3$ no contiene o cubre a $\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$ y en consecuencia el elemento matricial m_{11} es 0.

Tomemos luego, el implicante 2 y el término producto 1)

$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$	01	01	01	01
$\bar{x}_3\bar{x}_4$	00	00	01	01
	01	01	01	01

y el resultado nos indica que el término producto está cubierto por el implicante primo, por lo tanto, $m_{21} = 1$

MATRIZ DE CUBRIMIENTO

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

fig. 12)

El siguiente paso previo a la obtención de la mínima expresión, es el de determinar cuales son los implicantes primos "esenciales". Si existe uno o más terminos producto de la expresión expandida que estén cubiertos por un solo implicante primo, este implicante será esencial puesto que no existe posibilidad de que sea reemplazado por otro implicante primo. Esta condición es visible en la matriz de cubrimiento de la fig. 12) cuando alguna columna contiene solamente un "1"; el implicante primo esencial será el correspondiente al de la fila en la que se encuentra ese "1". Las columnas 1), 3), y 5) contienen solamente un "1" en las filas 2), 3), y 1) respectivamente; en consecuencia los implicantes correspondientes a las filas anteriormente mencionadas serán esenciales.

Nuestro paso final será seleccionar grupos de implicantes primos, en los cuales estén involucrados todos los implicantes primos esenciales y que cumplan con el requisito de que al sumar las filas correspondientes al grupo escogido, en la matriz de cubrimiento se obtenga un vector cubierto de "unos". Si esta restricción no se cumple, entonces aquel grupo no formará la mínima expresión de la función tratada y por lo tanto se deberá escoger otro.

METODO DE PERLOWSKI

Este método, está basado en el consensus iterado. La diferencia principal con el método de Mott radica en la forma de obtención de la mínima expresión de la función, a partir del conjunto de implicantes primos. En efecto, como hemos visto, Mott combina el consensus con el método tabular, mientras Perlowski aplica la teoría del consensus hasta alcanzar la mínima cobertura. Por esta razón nuestra atención va a dirigirse hacia el problema de la mínima cobertura de la función, suponiendo que contamos con el conjunto de implicantes primos previamente obtenido por aplicación del consensus iterado.

Sea la función a minimizar:

$$F = x_1x_2\bar{x}_3x_4 + x_1x_2\bar{x}_3\bar{x}_4 + x_1x_2x_3\bar{x}_4 + x_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3x_4$$

cuya tabla de implicantes primos es la siguiente:

TABLA DE IMPLICANTES PRIMOS

	x_1	x_2	x_3	x_4
$x_1x_2\bar{x}_3$	10	10	01	00
$x_1x_3\bar{x}_4$	10	00	10	01
$x_2\bar{x}_3x_4$	00	10	01	10
$x_1x_2\bar{x}_4$	10	10	00	01

fig. 13)

Con los datos de la fig. 13), procedemos a determinar la mínima expresión de la función dada. Para ello seguimos los siguientes pasos:

1.- Determinar el conjunto de implicantes primos esenciales Esto se lleva a cabo, separando un implicante primo del resto de implicantes y aplicando la siguiente regla: El implicante primo separado es esencial si no es generado por aplicación del consensus iterado al resto de implicantes. Repitiendo el procedimiento con todos y cada uno de los implicantes primos se obtiene el conjunto de implicantes primos esenciales. Para aplicar este primer paso al ejemplo, separemos el implicante $(x_1x_2\bar{x}_3)$ de la fig.13) y realicemos la operación de consensus con los implicantes $(x_2\bar{x}_3x_4)$ y $(x_1x_2\bar{x}_4)$ lo que nos da como resultado, $x_1x_2\bar{x}_3$, con lo que hemos generado el término previamente separado, hecho que significa que éste no es implicante primo esencial. Ahora separamos del grupo al término $(x_1x_3\bar{x}_4)$ y sobre el grupo restante aplicamos la operación de consensus; esto solo es posible con $(x_2\bar{x}_3x_4)$ y $(x_1x_2\bar{x}_4)$, produciéndose el término $(x_1x_2\bar{x}_3)$; esto nos indica que el término separado, $(x_1x_3\bar{x}_4)$, es esencial. De igual manera encontramos que el término $(x_2\bar{x}_3x_4)$ es un implicante primo esencial.

Veamos si los resultados anteriores son esperados mediante el método de Mott. En la fig. 14) consta la matriz de cubrimiento de la función del ejemplo; es facil visualizar en ella que el segundo y tercer implicante constituyen implicantes primos esenciales, resultado que coincide con el encontrado por el método de Perlowski

2.- Determinar los <u>implicantes primos eliminables</u>

Son eliminables los implicantes primos que no son esenciales. Por lo tanto, basta con separar los implicantes primos esenciales del conjunto de implicantes primos, para obtener los eliminables. En el ejemplo, $(x_1x_2\overline{x}_3)$ y $(x_1x_2\overline{x}_4)$ son eliminables.

3.- Determinar los <u>implicantes absolutamente eliminables</u>

Un procedimiento del consensus iterado sobre el conjunto de implicantes primos esenciales, nos indica cuales de los implicantes primos eliminables son absolutamente eliminables. Si la operación de consensus nos genera un implicante eliminable, luego este será absolutamente eliminable, pues estará contenido en el conjunto de esenciales. Los implicantes $(x_1x_3\overline{x}_4)$ y $(x_2\overline{x}_3x_4)$ del ejemplo, son esenciales y sobre estos no es posible aplicar la operación de consensus lo que nos indica que ninguno de los implicantes eliminables es absolutamente eliminable.

4.- Eliminar los implicantes primos absolutamente eliminables

De esta manera la expresión de la función estará dada por los implicantes primos esenciales y los condicionalmente eliminables.

5.- Obtener la <u>expresión mínima</u>

a). Tomar el conjunto de implicantes primos esenciales más un eliminable y sobre este conjunto aplicar el

MATRIZ DE CUBRIMIENTO

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & \underline{1} & 0 \\ 1 & 0 & 0 & 0 & \underline{1} \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

TERMINOS PRODUCTO

1) $x_1x_2\bar{x}_3x_4$

2) $x_1x_2\bar{x}_3\bar{x}_4$

3) $x_1x_2x_3\bar{x}_4$

4) $x_1\bar{x}_2x_3\bar{x}_4$

5) $\bar{x}_1x_2\bar{x}_3x_4$

IMPLICANTES PRIMOS

1) $x_1x_2\bar{x}_3$

2) $x_1x_3\bar{x}_4$

3) $x_2\bar{x}_3x_4$

4) $x_1x_2\bar{x}_4$

fig. 14)

consensus iterado. Si todos los implicantes primos son obtenidos de esta operación, el conjunto tomado es una expresión mínima de la función. El procedimiento se repite tomando cada vez un implicante eliminable y el conjunto esencial; esto es necesario puesto que la función puede tener una o más expresiones con igual número de términos. Si no se ha obtenido todos los implicantes primos de la operación de consensus entonces el procedimiento se continúa.

b). Se realiza el mismo paso anterior pero esta vez tomando dos implicantes primos eliminables. Si aun así no se encuentra todos los implicantes primos eliminables de la operación de consensus, será necesario tomar tres implicantes primos eliminables y en algunos casos hasta cuatro. Generalmente siguiendo este procedimiento se llega a obtener una expresión mínima.

Para aplicar esta parte del método en nuestro ejemplo, tomemos los dos implicantes primos esenciales, $(x_1x_3\overline{x}_4)$ y $(x_2\overline{x}_3x_4)$ más el término eliminable $(x_1x_2\overline{x}_3)$. De este grupo, con los términos $(x_1x_2\overline{x}_3)$ y $(x_1x_3\overline{x}_4)$, realizamos la operación de consensus, lo que nos da como resultado $(x_1x_2\overline{x}_4)$, que es el otro implicante primo eliminable, razón por la cual el grupo que hemos escogido constituye ya una mínima expresión. Tomemos ahora el grupo formado por $(x_1x_3\overline{x}_4)$, $(x_2\overline{x}_3x_4)$ $(x_1x_2\overline{x}_4)$; un consensus entre estos dos últimos nos restituye $(x_1x_2\overline{x}_3)$, el implicante eliminable sobrante, y en consecuencia este grupo elegi do es también una expresión mínima. La función nos quedaría reducida a cualquiera de las dos expresiones siguientes:

$$F = x_1x_3\overline{x}_4 + x_2\overline{x}_3x_4 + x_1x_2\overline{x}_3 \quad (1)$$

$$F = x_1x_3\overline{x}_4 + x_2\overline{x}_3x_4 + x_1x_2\overline{x}_4 \quad (2)$$

el procedimiento termina aquí.

ESTUDIO COMPARATIVO DE LOS PRINCIPALES METODOS

Como hemos visto en el capítulo anterior, el método del Mapa de Karnaugh, es un método visual bastante eficaz, siempre y cuando se cuente con funciones de no más de cuatro variables, lo que constituye una seria limitación. Podemos anotar otra desventaja y es el hecho de que nada nos dice este método sobre la mínima expresión de la función, puesto que solo nos genera el conjunto de implicantes primos, siendo preciso recurrir a otros métodos para resolver el problema de la mínima cobertura.

El uso de las condiciones no implementadoras o indefinidad en el método de Karnaugh, resulta ser una característica interesante del mismo.

El método tabular, al igual que el de Karnaugh, usa las condiciones no implementadoras para la obtención de la tabla de implicantes primos, pero presenta la ventaja de que no está limitada por el número de variables, al menos si de programación se trata.

Cuando se intenta programar este método se aprecia su adaptabilidad al lenguaje de máquina y su sencillez de algoritmización. Sin embargo, puede acarrear dificultades si no se cuenta con un computador de mediana o gran capacidad de memoria. Pese a sus limitaciones podemos considerar a este método como útil, eficaz y versatil, que en la generalidad de los casos nos lleva a soluciones mínimas muy aceptables.

El método de Mott reune las cualidades y ventajas del método tabular, sin embargo, el método en si mismo es menos directo y cuando se lo programa se encuentra mayor dificultad. Una característica por la cual el método de Mott aventaja al de Quine-MacCluskey, es el hecho de que la función no necesariamente debe estar en su forma canónica, lo cual en algunas ocasiones resulta muy ventajoso por el ahorro de tiempo y trabajo demandados al expandir la función a su forma canónica. Al respecto hay que aclarar lo siguiente: el método de Mott, como se ha visto en el capítulo anterior, requiere para la obtención de la mínima expresión que la función esté en su forma canónica; razón por la cual si el método es realizado a mano no existe tal ventaja, pero cuando el método está ya programado esta ventaja se hace realidad puesto que el programa se encargará de realizar la expansión.

El método de Mott combina el consensus iterado con la tabulación. Perlowski en su método elimina esta mescla, pues encuentra la mínima expresión aplicando en forma reiterada el principio del consensus. Si el método de Perlowski es programado se logra ahorrar tiempo y memoria de computación ya que una misma subrutina puede ser utilizada, tanto para la determinación de los implicantes primos, como de la mínima expresión. Aun así el método de Mott es preferido en razón de que utili za las condiciones no implementadoras, por lo cual los resultados son más òptimos cuando el problema a tratarse contiene dichas condiciones.

De lo anterior puede deducirse que los métodos más representativos y utilizados son el método tabular o de Quine-MacCluskey y el método de Mott. También en este trabajo han sido preferidos y escogidos para una ulterior programación.

A partir de los resultados de cualquiera de estos dos métodos se procederá a encontrar una solución del problema en lógica NAND de hasta tres niveles.

MINIMIZACION EN LOGICA NAND

En el presente capítulo se estudiarán métodos de simplificación en Lógica NAND propiamente dicha.

METODO DE DIETMEYER-SU,

Este es un método utilizado cuando existe restricción del número de entradas por compuerta (Fan-in limitado). Consiste en la búsqueda de un factor común que cumpla además con ciertos requisitos. En general estos requisitos se reducen al encuentro de una máxima figura de merito.

Los autores presentan para la resolución del problema, tres algoritmos y algunas sofisticaciones de los mismos. El algoritmo a usarse depende del problema y los tres pueden o no dar resultados diferentes.

El método es sumamente complejo y las subrutinas que los autores dan como referencia estan compilados en una obra de difícil consecución en nuestro País. Por estas razones y por el hecho de que el método solo es útil cuando el número de variables es grande, se ha desistido de una programación del mismo.

METODO DE MUROGA-IBARAKI

El método, capaz de resolver problemas de minimización tanto en Lógica NAND, NOR, o Lógica Combinada, tiene como puento de partida un código de <u>Programación Entera</u> denominado "ILLIP"

(Illinois Integer Programming Code) y que está basado en la enumeración implicita, consiste en la transformación del problema de minimización a un problema de programación lineal entera y como tal es luego implementado y resuelto. Al parecer este método es bastante versatil ya que puede resolver problemas con cualquier tipo de restricción.

Este método no se programó por dos razones:

1.- Como ya se anotó anteriormente, el método precisa de una codificación especial, por lo que se requiere de información adicional, cuya consecución se lograría en un tiempo no razonable.

2.- Resolver una programación lineal entera está fuera del alcance y perspectivas de este trabajo.

Considero que un estudio ampliado puede ser de interés y utilidad.

TRANSFORMACION DE LOGICA AND-OR A LOGICA NAND

Una de las reglas básicas del Algebra Booleana denominada "Teorema de Morgan" nos ayuda a transformar una expresión de Lógica AND-OR a Lógica NAND. En efecto estos teoremas vienen expresados por:

1) $\overline{X + Y} = \overline{X}\,\overline{Y}$

2) $\overline{X\,Y} = \overline{X} + \overline{Y}$

donde X y Y son literales o expresiones booleanas.

Si a la expresión 1) la complementamos en sus dos lados tenemos:

$$\overline{\overline{X + Y}} = \overline{\bar{X}\,\bar{Y}}$$

lo cual da como resultado:

$$X + Y = \overline{\bar{X}\,\bar{Y}}$$

Con lo que se ha logrado la transformación deseada.

Ejemplo:

Sea $F = x_1x_4 + \bar{x}_1\bar{x}_2\bar{x}_4 + x_3x_4 + x_1\bar{x}_3$

la función que deseamos transformar, entonces;

$$F = \overline{\overline{x_1x_4} + \overline{\bar{x}_1\bar{x}_2\bar{x}_4} + \overline{x_3x_4} + \overline{x_1\bar{x}_3}}$$

será la expresión de F en lógica NAND

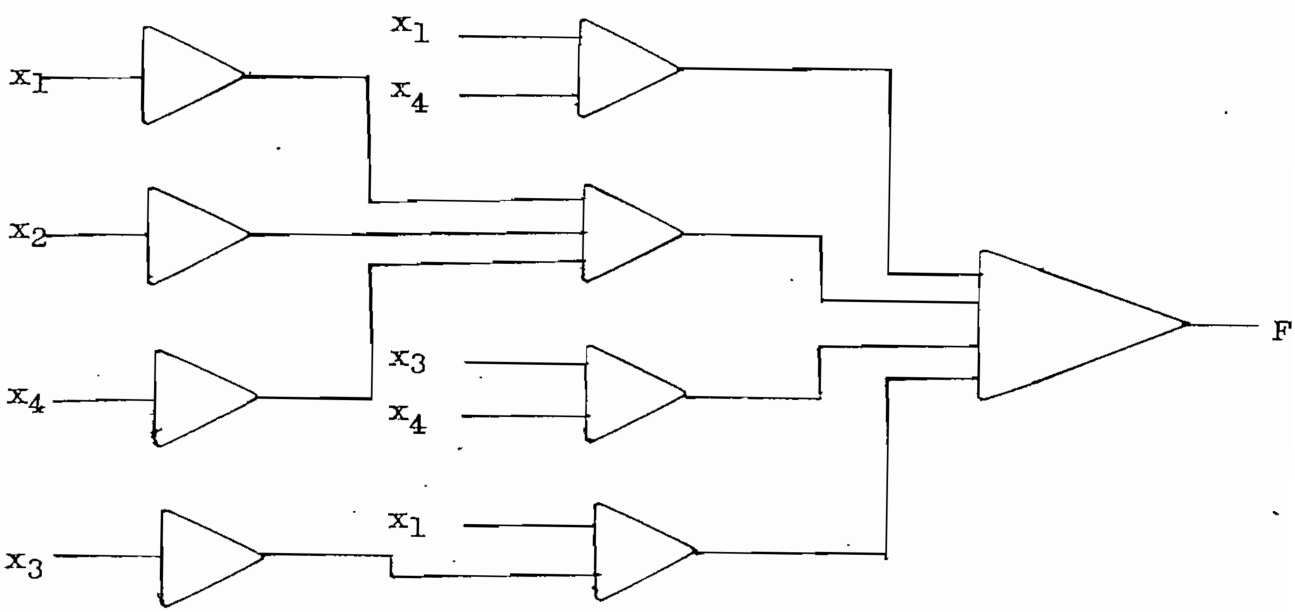


fig. 15)

De lo anterior podemos deducir las siguientes reglas para la transformación AND-OR a NAND:

1) Invertir todos los productos
2) Transformar las sumas a productos
3) Invertir la expresión resultante de los pasos anteriores

Las reglas anteriores pueden reducirse a remplazar las compuertas AND-OR por compuertas NAND.
Este tipo de transformación es válida cuando no se tiene ningun tipo de restricción.

Podemos aplicar esta transformación a una función minimizada por cualquiera de los métodos estudiados en el capítulo anterior. El resultado que obtendríamos sería una implementación mínima o cuasi mínima en Lógica NAND de hasta tres niveles, considerando un nivel más debido a la inversión en las entradas.

Esta transformación la vamos a utilizar en nuestro problema de Minimización en Lógica NAND, ya que resulta sumamente sencillo y no precisa de una variación o ampliación de los programas para minimización de la Función Booleana.

PROGRAMAS

Programa correspondiente al Método de Quine-MacCluskey

Para cumplir con los objetivos de este trabajo se programó el Método de Quine-Mac-Cluskey en FORTRAM IV 360/370.

En este programa, las matrices usadas no son significativas, pues pierden su información inicial y su uso cambia de acuerdo a las necesidades del programa.

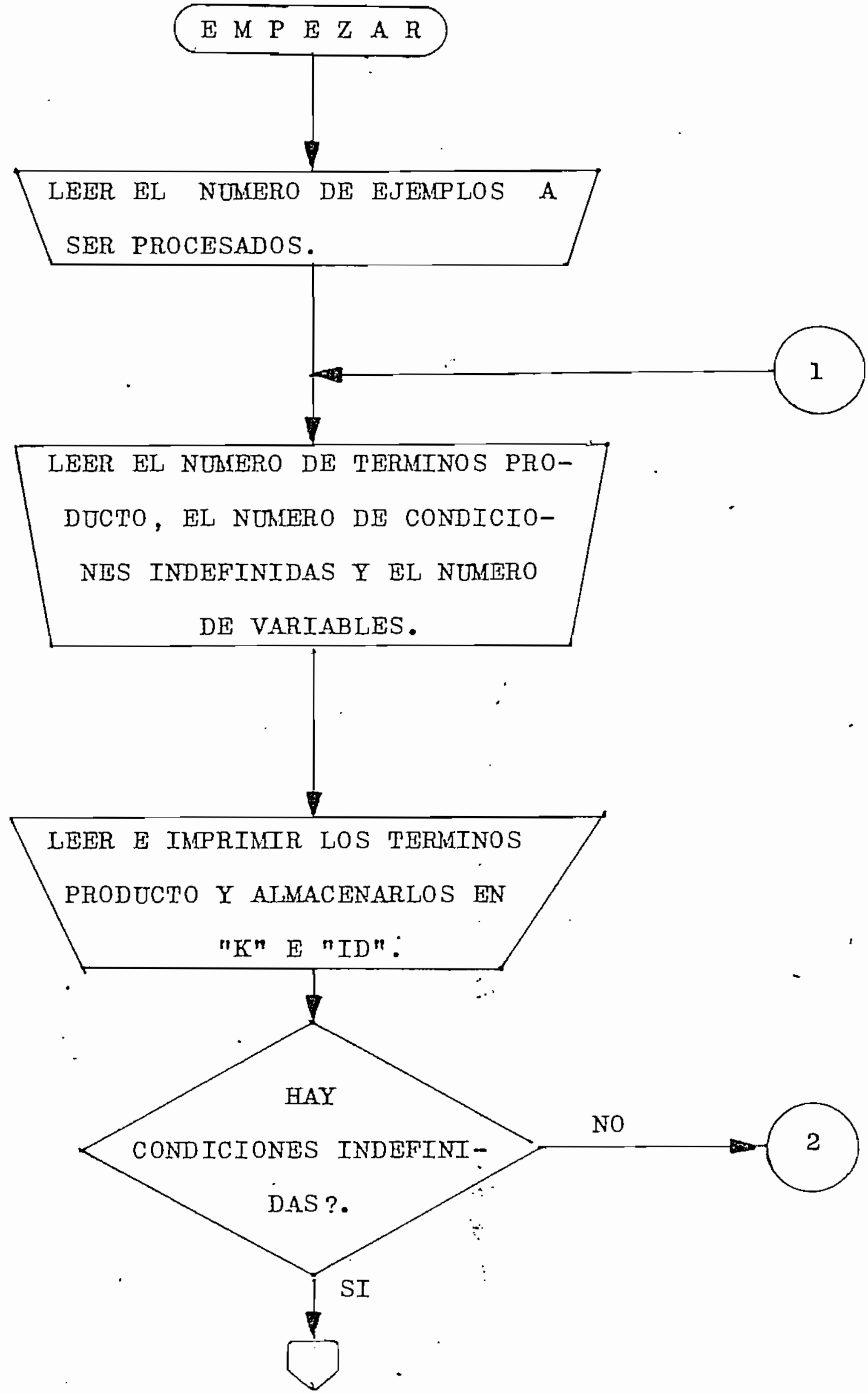
La Matriz "K" almacena en un principio, los elementos correspondientes a la función a ser minimizada. Las condiciones indefinidas tambien son guardadas en esta Matriz.

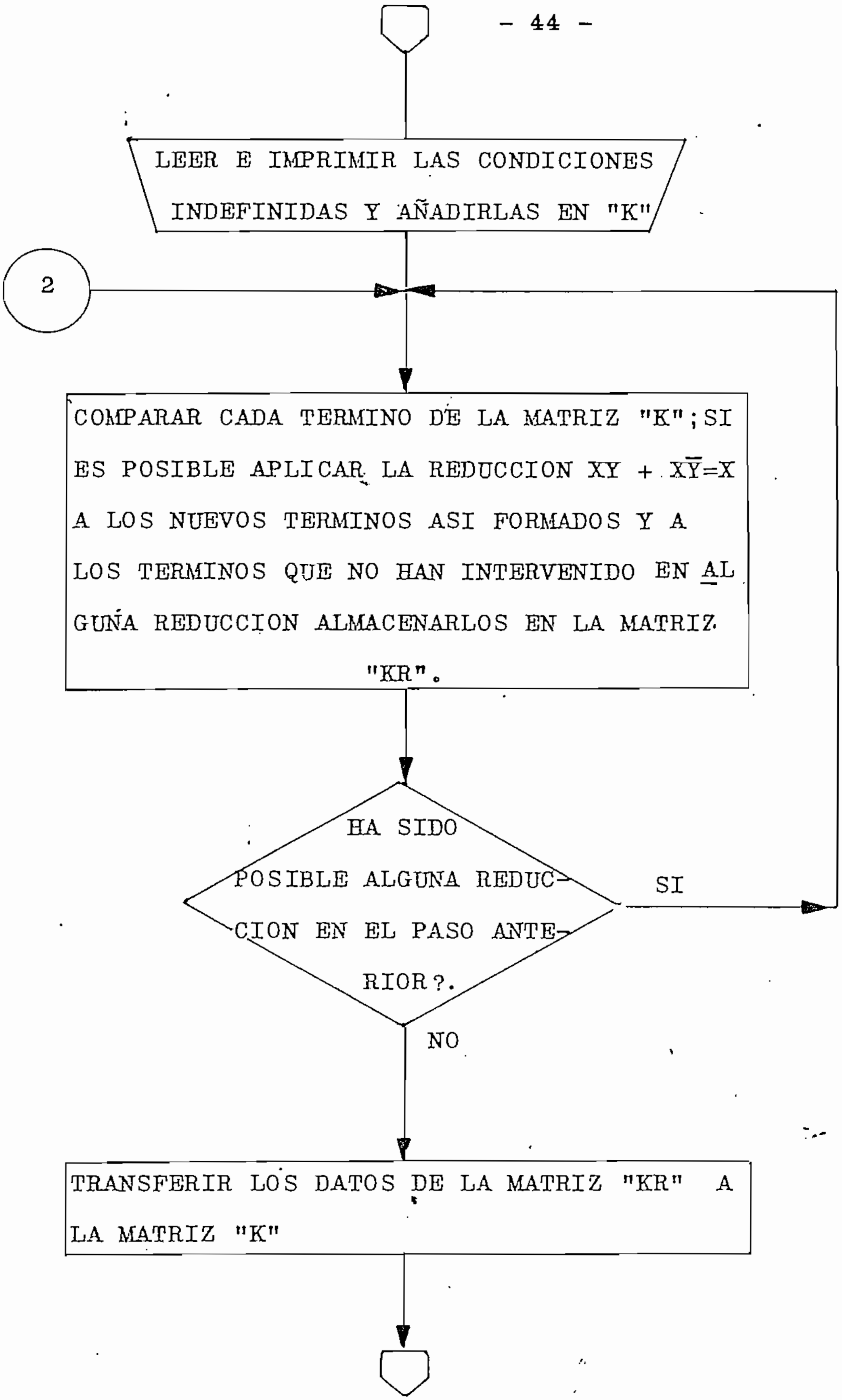
La Matriz "K" junto con la Matriz "KR" y el Vector "NP" se utiliza para la obtención de los implicantes primos.

La Matriz "ICHOI" esta encargada de contener los datos de las Tablas de Selección. Para la formación de estas tablas de selección se utiliza la Matriz "K"

Para la selección de los Implicantes Primos Esenciales y Elegibles se utilizan las Matrices "K", "KR", "ICHOI" y "KES".

Las dimensiones correspondientes a las matrices y a los vectores constan en el listado del programa.


E M P E Z A R
LEER EL NUMERO DE EJEMPLOS A SER PROCESADOS.
1
LEER EL NUMERO DE TERMINOS PRODUCTO, EL NUMERO DE CONDICIONES INDEFINIDAS Y EL NUMERO DE VARIABLES.
LEER E IMPRIMIR LOS TERMINOS PRODUCTO Y ALMACENARLOS EN "K" E "ID".
HAY CONDICIONES INDEFINIDAS?.
NO
2
SI


LEER E IMPRIMIR LAS CONDICIONES INDEFINIDAS Y AÑADIRLAS EN "K"
2
COMPARAR CADA TERMINO DE LA MATRIZ "K"; SI ES POSIBLE APLICAR LA REDUCCION XY + X$\overline{Y}$=X A LOS NUEVOS TERMINOS ASI FORMADOS Y A LOS TERMINOS QUE NO HAN INTERVENIDO EN ALGUNA REDUCCION ALMACENARLOS EN LA MATRIZ "KR".
HA SIDO POSIBLE ALGUNA REDUCCION EN EL PASO ANTERIOR?.
SI
NO
TRANSFERIR LOS DATOS DE LA MATRIZ "KR" A LA MATRIZ "K"

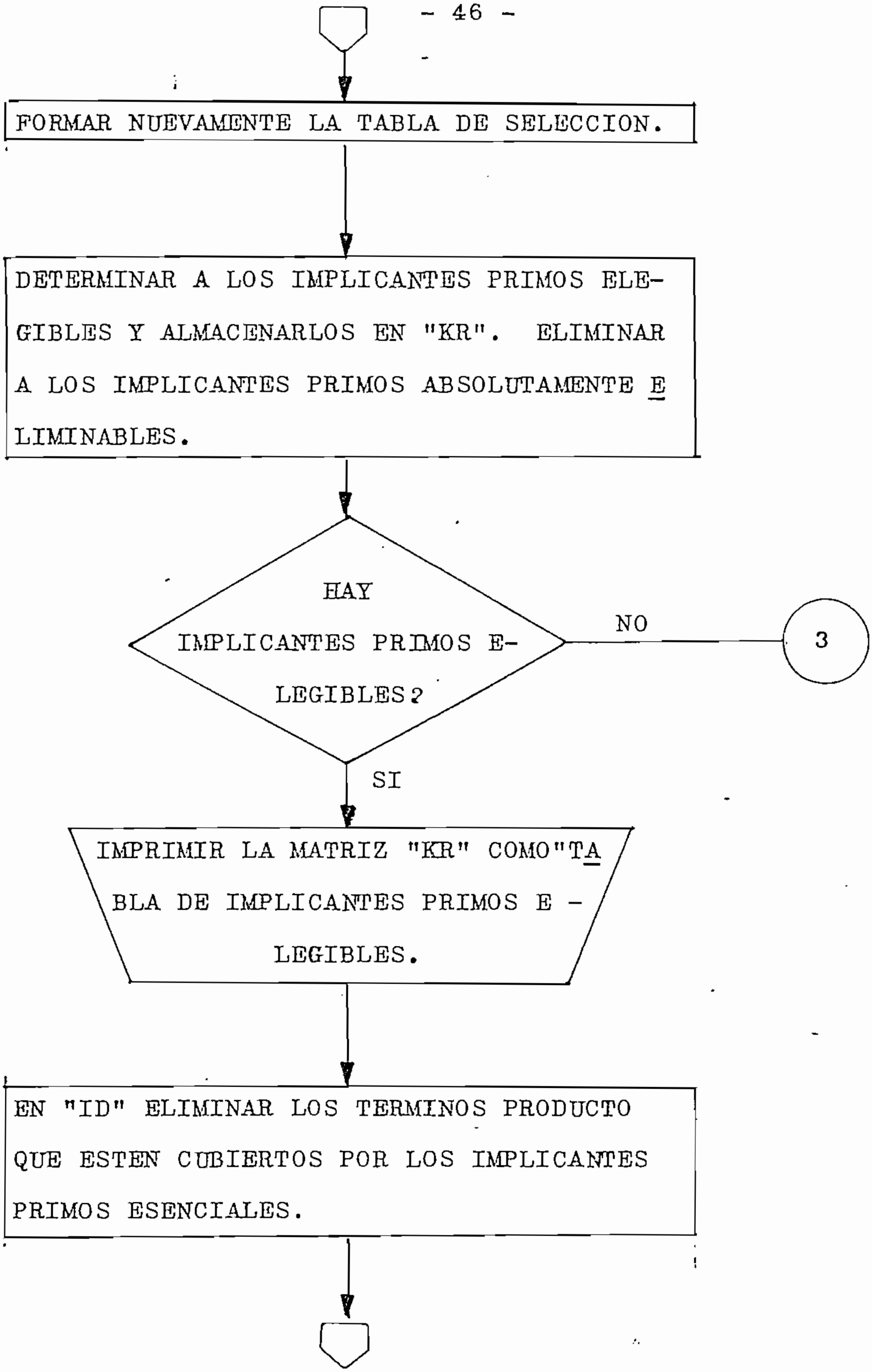
IMPRIMIR DATOS DE LA MATRIZ"KR" EN "TABLA DE IMPLICANTES PRIMOS".

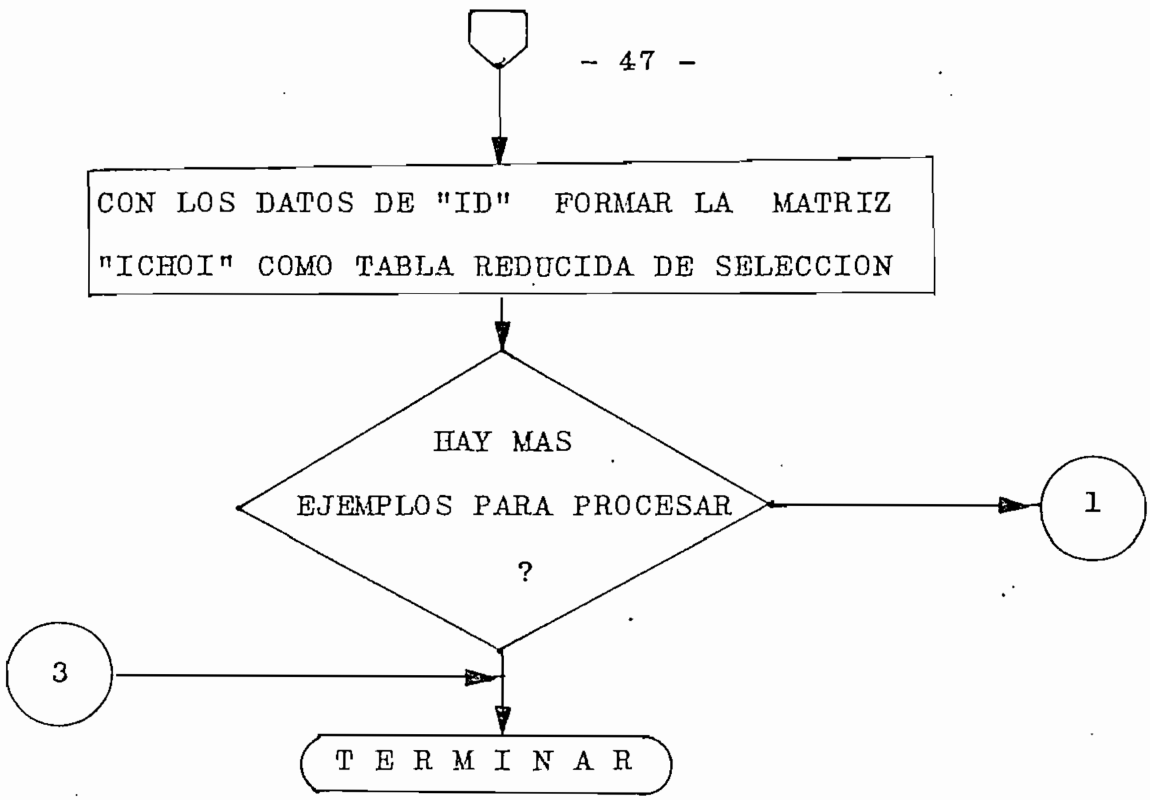
DETERMINANDO CUALES TERMINOS DE LA MATRIZ "KR" CUBREN A LOS TERMINOS DE "ID" , FORMAR EN LA MATRIZ "ICHOI" LA TABLA DE SELECCION.

DETERMINAR CUALES SON LOS IMPLICANTES PRIMOS ESENCIALES EN "KR" Y TRANSFERIRLOS A "K".

IMPRIMIR LA MATRIZ "K" COMO "TABLA DE IMPLICANTES PRIMOS ESENCIALES".

TRANSFERIR LOS TERMINOS NO ESENCIALES DE "KR" A "K". LOS TERMINOS ASI REORDENADOS TRANSFERIRLOS NUEVAMENTE A "KR".


FORMAR NUEVAMENTE LA TABLA DE SELECCION.
DETERMINAR A LOS IMPLICANTES PRIMOS ELEGIBLES Y ALMACENARLOS EN "KR". ELIMINAR A LOS IMPLICANTES PRIMOS ABSOLUTAMENTE ELIMINABLES.
HAY IMPLICANTES PRIMOS ELEGIBLES?
NO
3
SI
IMPRIMIR LA MATRIZ "KR" COMO "TABLA DE IMPLICANTES PRIMOS ELEGIBLES.
EN "ID" ELIMINAR LOS TERMINOS PRODUCTO QUE ESTEN CUBIERTOS POR LOS IMPLICANTES PRIMOS ESENCIALES.



CON LOS DATOS DE "ID" FORMAR LA MATRIZ "ICHOI" COMO TABLA REDUCIDA DE SELECCION
HAY MAS EJEMPLOS PARA PROCESAR ?
1
3
T E R M I N A R

GUIA DEL USUARIO DEL PROGRAMA CORRESPONDIENTE AL METODO DE QUINE MACCLUSKEY

El usuario de este programa debe atenerse a las siguientes instrucciones:

- La primera tarjeta de datos llevará perforado en las columnas 1 y 2 el número de ejemplos a ser procesados.
- A continuación vendrán grupos de tarjetas correspondientes a cada ejemplo a procesar. Estos grupos estarán compuestos

de una tarjeta que en las columnas 1 y 2 lleve el número de variables de la función; en las columnas 3 y 4 el número de productos que hacen la función igual a 1; en las columnas 5 y 6 el número de condiciones indefinidas; luego vendrá la(s) tarjeta(s) que con formato 80A1 contendrán los datos correspondientes a los términos producto de acuerdo al siguiente código:

- "1" representa una variable afirmada
- " 0" representa una variable negada
- "*" representa una variable ausente

Completando cada grupo irá la(s) tarjeta(s) que contenga(n) los datos correspondientes a las condiciones "indefinidas". Estas tarjetas estarán perforadas de acuerdo al formato y código indicados.

Ejemplo:

Sea $F = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2x_3 + x_1x_2\bar{x}_3 + x_1x_2x_3$

con las siguientes condiciones indefinidas:

$$\overline{x}_1 x_2 \overline{x}_3 \; + \; x_1 \overline{x}_2 x_3$$

La primera tarjeta llevaría en la columna 1 y 2, 01; la segunda tarjeta llevaría en la columna 1 y 2, 04; en las columnas 3 y 4, 05; en las columnas 5 y 6, 02. La tarjeta Nº 3 llevaría desde la columna 1 lo siguiente:

0000010111110111

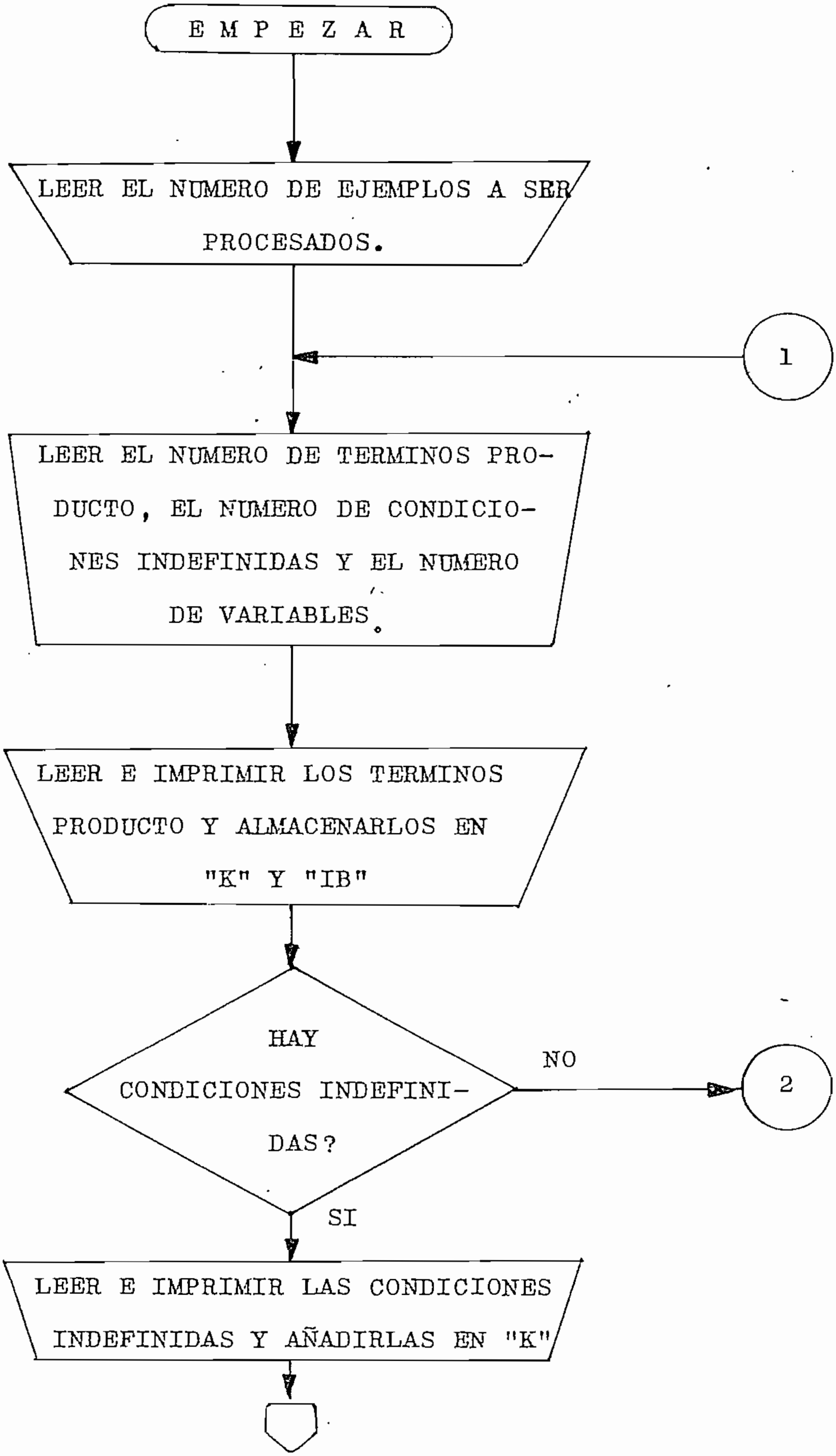
La cuarta tarjeta llevaría a partir de la columna 1 lo siguiente:

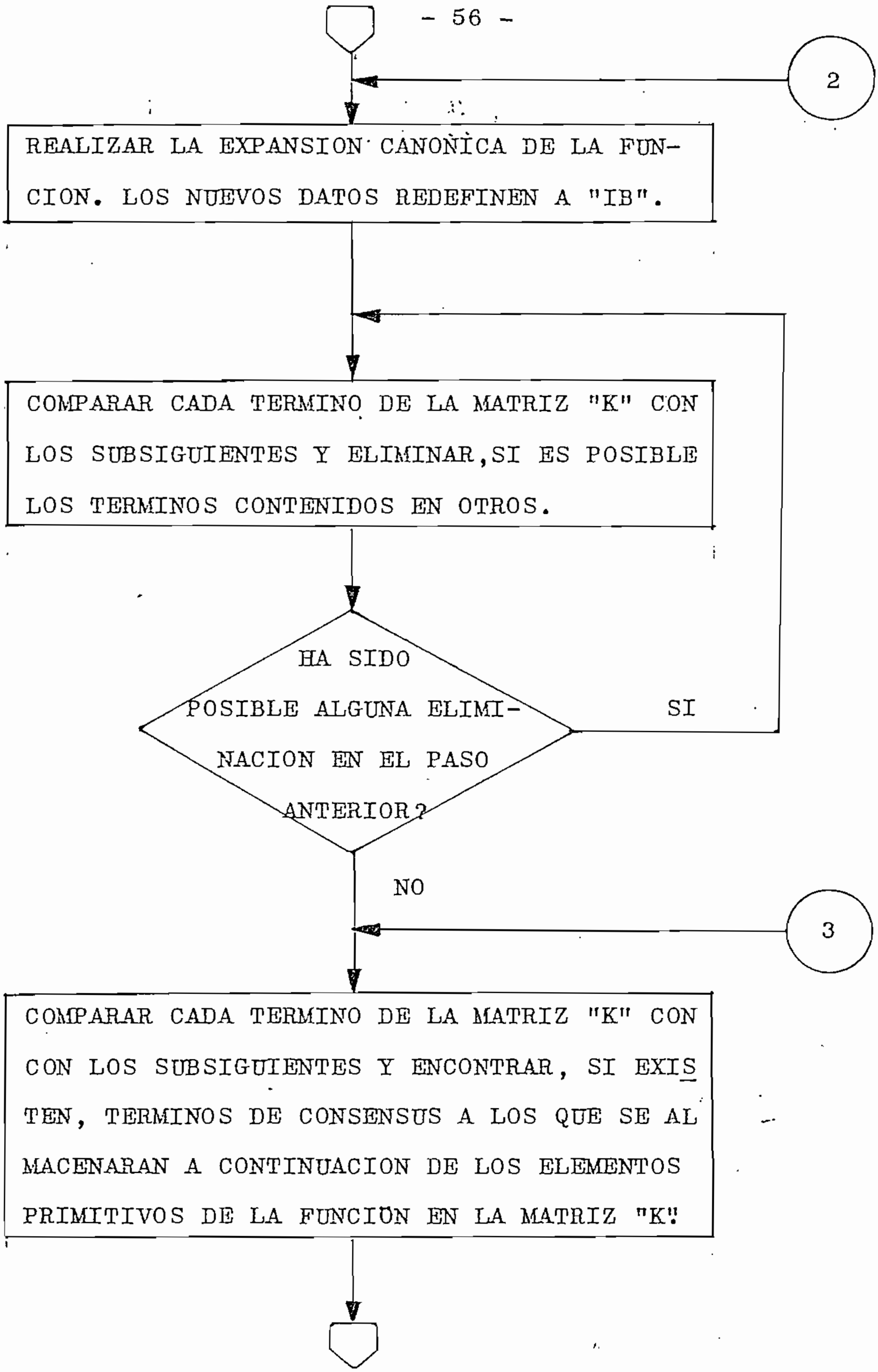
010101

El programa nos dará como resultado lo siguiente:

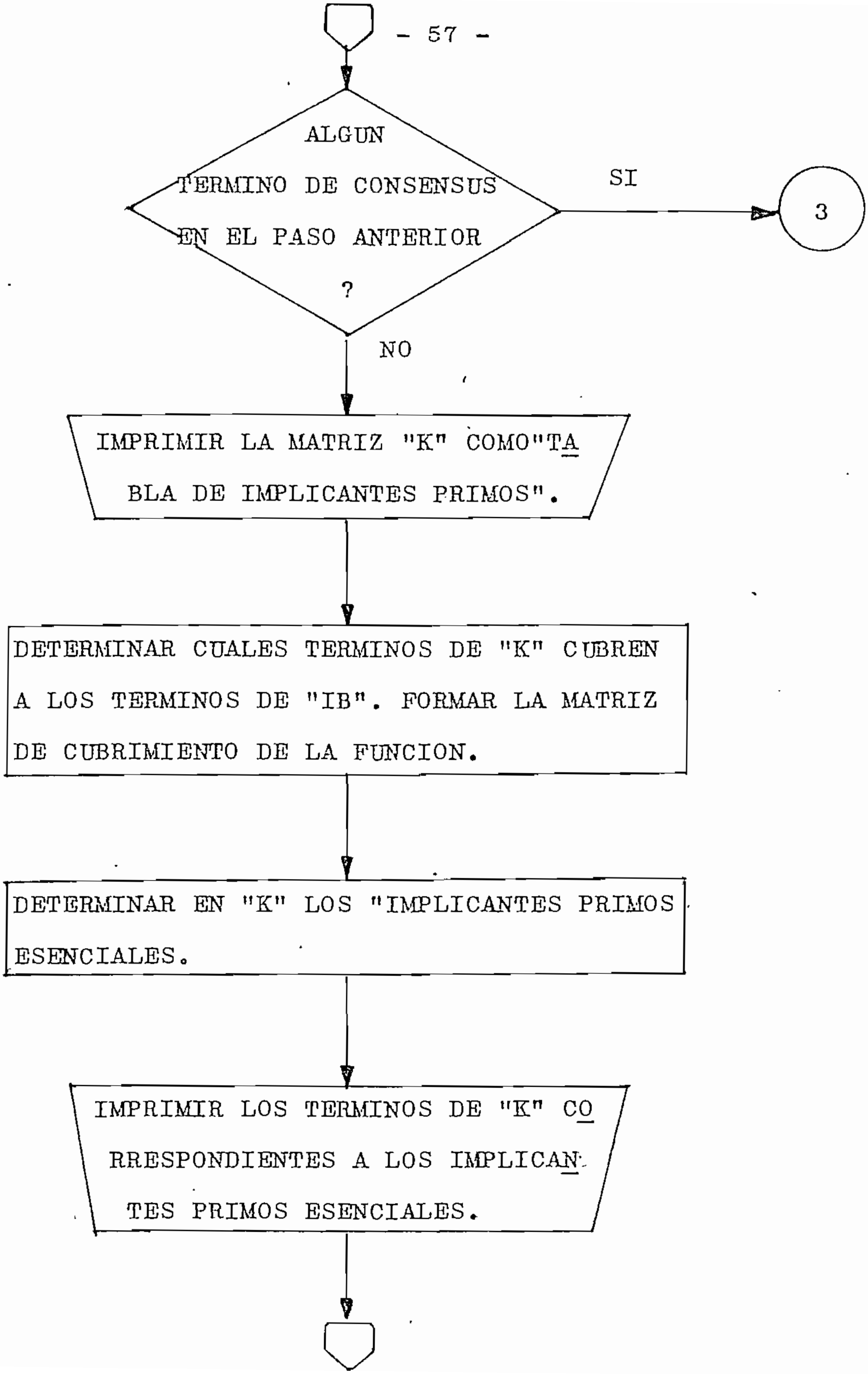
1) Tabla de Implicantes Primos
2) Tabla de Implicantes Primos Esenciales
3) Tabla de Implicantes Primos Elegibles, solamente si existe
4) Tabla Reducida de Selección, si existe Implicantes Primos Elegibles

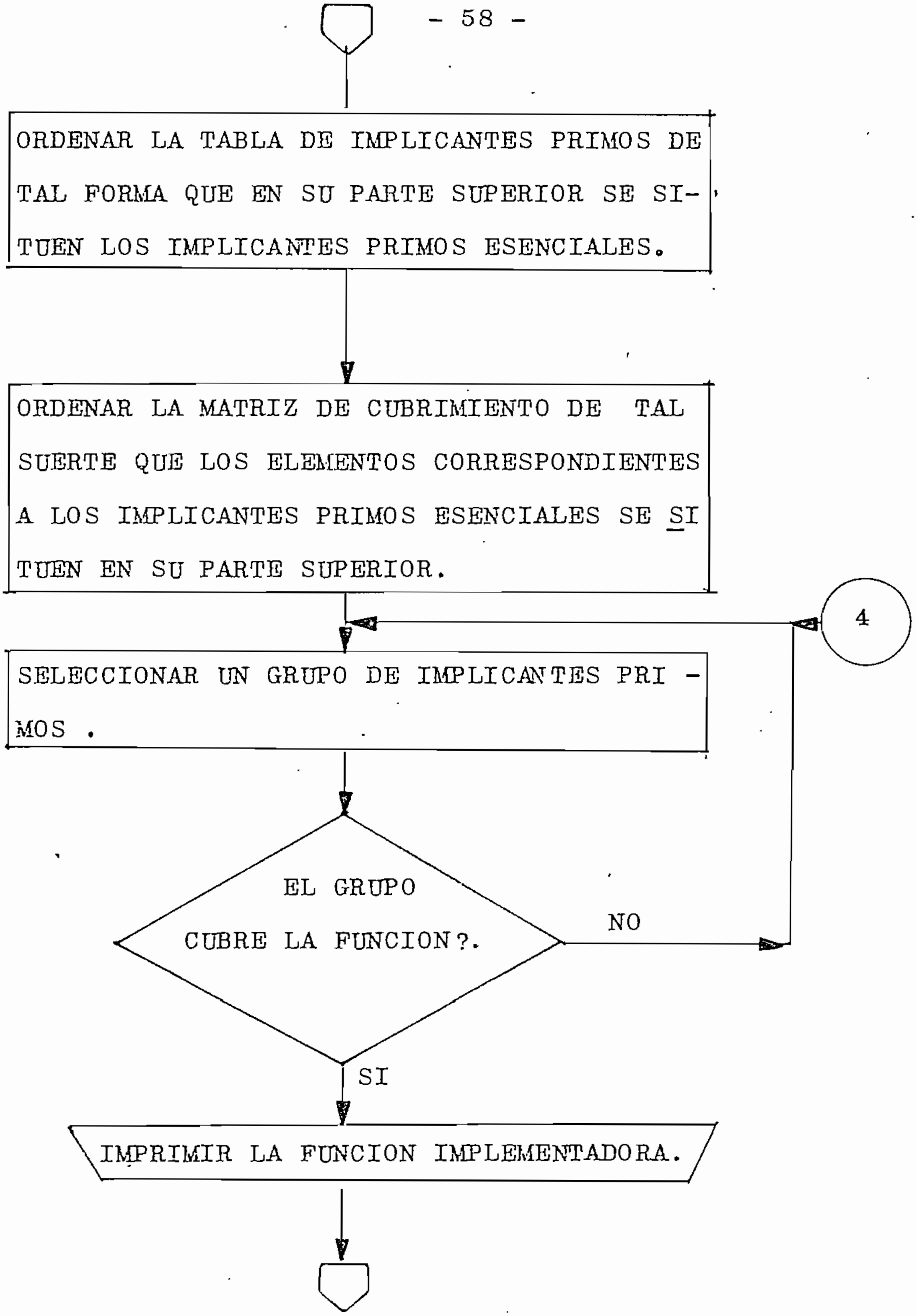
Como puede observarse, este programa no nos dá directamente la(s) mínima(s) expresion(es); en su lugar la tabla reducida de selección impresa nos permitirá encontrar la mínima expresión de acuerdo con lo estudiado en capítulos previos.

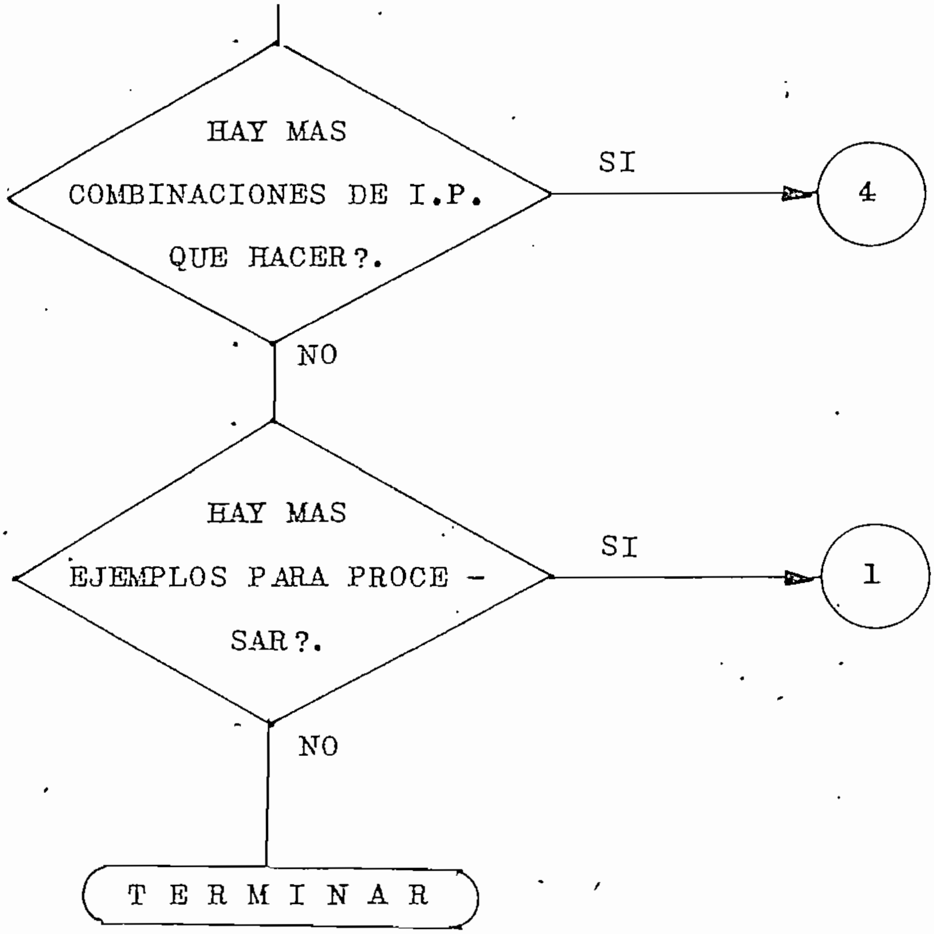

E M P E Z A R
LEER EL NUMERO DE EJEMPLOS A SER PROCESADOS.
1
LEER EL NUMERO DE TERMINOS PRODUCTO, EL NUMERO DE CONDICIONES INDEFINIDAS Y EL NUMERO DE VARIABLES.
LEER E IMPRIMIR LOS TERMINOS PRODUCTO Y ALMACENARLOS EN "K" Y "IB"
HAY CONDICIONES INDEFINIDAS?
NO
2
SI
LEER E IMPRIMIR LAS CONDICIONES INDEFINIDAS Y AÑADIRLAS EN "K"



2
REALIZAR LA EXPANSION CANONICA DE LA FUNCION. LOS NUEVOS DATOS REDEFINEN A "IB".
COMPARAR CADA TERMINO DE LA MATRIZ "K" CON LOS SUBSIGUIENTES Y ELIMINAR,SI ES POSIBLE LOS TERMINOS CONTENIDOS EN OTROS.
HA SIDO POSIBLE ALGUNA ELIMINACION EN EL PASO ANTERIOR?
SI
NO
3
COMPARAR CADA TERMINO DE LA MATRIZ "K" CON CON LOS SUBSIGUIENTES Y ENCONTRAR, SI EXISTEN, TERMINOS DE CONSENSUS A LOS QUE SE ALMACENARAN A CONTINUACION DE LOS ELEMENTOS PRIMITIVOS DE LA FUNCION EN LA MATRIZ "K".


ALGUN
TERMINO DE CONSENSUS
EN EL PASO ANTERIOR
?
SI
3
NO
IMPRIMIR LA MATRIZ "K" COMO"TABLA DE IMPLICANTES PRIMOS".
DETERMINAR CUALES TERMINOS DE "K" CUBREN A LOS TERMINOS DE "IB". FORMAR LA MATRIZ DE CUBRIMIENTO DE LA FUNCION.
DETERMINAR EN "K" LOS "IMPLICANTES PRIMOS ESENCIALES.
IMPRIMIR LOS TERMINOS DE "K" CORRESPONDIENTES A LOS IMPLICANTES PRIMOS ESENCIALES.


ORDENAR LA TABLA DE IMPLICANTES PRIMOS DE TAL FORMA QUE EN SU PARTE SUPERIOR SE SITUEN LOS IMPLICANTES PRIMOS ESENCIALES.
ORDENAR LA MATRIZ DE CUBRIMIENTO DE TAL SUERTE QUE LOS ELEMENTOS CORRESPONDIENTES A LOS IMPLICANTES PRIMOS ESENCIALES SE SITUEN EN SU PARTE SUPERIOR.
4
SELECCIONAR UN GRUPO DE IMPLICANTES PRIMOS.
EL GRUPO CUBRE LA FUNCION?.
NO
SI
IMPRIMIR LA FUNCION IMPLEMENTADORA.


HAY MAS
COMBINACIONES DE I.P.
QUE HACER?.
SI
4
NO
HAY MAS
EJEMPLOS PARA PROCE -
SAR?.
SI
1
NO
T E R M I N A R

GUIA DEL USUARIO DEL PROGRAMA CORRESPONDIENTE AL METODO DE MOTT

El usuario de este programa debe atenerse a las siguientes instrucciones:

- La primera tarjeta de datos llevará perforada en las columnas 1 y 2 el número de ejemplos a ser procesados.
- A continuación vendrán grupos de tarjetas correspondientes a cada ejemplo. Estos grupos estarán compuestos de una tarjeta que en las columnas 1 y 2 lleve el número de productos implementadores; en las columnas 3 y 4, el número de productos de condición indefinida; en las columnas 5 y 6 el número de variables. Luego vendrá la(s) tarjeta(s), que con formato 40A2 contendrá los datos correspondientes a los productos implementadores, de acuerdo al siguiente código:

- "10" representa una variable afirmada
- "01" representa una variable negada
- "00" representa una variable ausente

Completando cada grupo de las tarjetas que contenga los datos correspondientes a cada ejemplo vendran tarjeta(s) que contenga(n) a los productos de condición indefinida; dichas tarjetas estarán perforadas de acuerdo al formato y código indicados.

Ejemplo:

Si $F'= \bar{x}_1\bar{x}_2x_3 + \bar{x}_1\bar{x}_2x_4 + x_2\bar{x}_3\bar{x}_4 + x_1x_2x_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3x_4$

y si los siguientes productos son de condición indefinida:

$$x_2x_3x_4 \quad + \quad x_1\overline{x}_2x_3\overline{x}_4$$

La primera tarjeta de datos llevaría en la columna 1 y 2, 01; la segunda tarjeta llevaría en las columnas 1 y 2, 05; en las columnas 3 y 4, 02; y en las columnas 5 y 6,02. La tarjeta Nº 3 llevaría desde la columna 1 lo siguiente:

0101100001010010001001011010100110010110

La cuarta tarjeta llevaría a partir de la columna 1 lo siguiente:

0010101010011001

El programa nos dará como resultado lo siguiente:

1) Tabla de Implicantes Primos
2) Tabla de Implicantes Primos Esenciales
3) Un grupo de funciones que implementen a la función original.

```
C        'PROGRAMA PARA MINIMIZAR UNA FUNCION BOOLEANA
C         POR EL METODO DE MOTT(CONSENSUS)
C
C
         DIMENSION KA(20,20),K(50,10),KB(20,20),IK(50),IB(20,2
         DIMENSION KM(30),NK(30)
         DATA K1,K2,K3,K4/'00','01','10','11'/
 100     FORMAT (3I2)
 101     FORMAT (40A2)
 102     FORMAT (10X,32I2)
 103     FORMAT (10X,A2,2X,A2,2X,A2,2X,A2,2X,A2,2X,A2,2X,A2//)
 105     FORMAT(///,2X,'TABLA DE IMPLICANTES PRIMOS ESENCIALES
        1/,1X,40('*'))
 106     FORMAT(///,2X,'TABLA DE IMPLICANTES PRIMOS',4X,
        1'METODO DE MOTT        ',/,1X,70('*'))
 107     FORMAT(///,2X,70('='),//,2X,'FUNCION IMPLEMENTADORA N
        1'MERO',I3,/,5X,35('='),///)
 108     FORMAT(1H1,1X,'FUNCION BOOLEANA A MINIMIZAR',/,6X,
        1'ESTA REPRESENTADA POR LA SUMA DE LOS SIGUIENTES ',
        1'PRODUCTOS',/,5X,50('='),///)
 109     FORMAT(//,10X,'LAS SIGUIENTES SON CONDICIONES ',
        1'INDEFINIDAS',/,5X,50('='),///)
         READ(1,100)NEJ
         DO 7000 NTR=1,NEJ
         READ(1,100)NI,ND,MJ
         IROZ=NI+1
         N =NI+ND
C
C  LEE E IMPRIME LA FUNCION A SER MINIMIZADA
C
         READ(1,101)((K(M,J),J=1,MJ),M=1,NI)
         WRITE(3,108)
         DO 200 NAL=1,NI
 200     WRITE(3,103)(K(NAL,J),J=1,MJ)
         DO 201 LU=1,NI
         DO 201 J=1,MJ
 201     IB(LU,J)=K(LU,J)
         IF(ND.EQ.0)GO TO 210
C
C  LEE E IMPRIME LAS CONDICIONES INDEFINIDAS
C
         READ(1,101)((K(M,J),J=1,MJ),M=IROZ,N)
         WRITE(3,109)
         DO 220 NAL=IROZ,N
 220     WRITE(3,103)(K(NAL,J),J=1,MJ)
C
C  REALIZA LA EXPANSION DE LA FUNCION A SU FORMA ESTANDAR
C
 210     NUD=NI
         ISA=NUD
 202     DO 211 LU=1,NUD
         DO 212 J=1,MJ
         IF(IB(LU,J).EQ.K1)GO TO 230
         GO TO 212
 230     IB(LU,J)=K2
         DO 204 KAS=1,MJ
 204     IB(ISA+1,KAS)=IB(LU,KAS)
         IB(ISA+1,J)=K3
         ISA=ISA+1
 212     CONTINUE
 211     CONTINUE
         IF(ISA.EQ.NUD)GO TO 215
 214     NUD=ISA
         GO TO 202
C
C SE INICIA EL CALCULO DE IMPLICANTES PRIMOS
C
 215     NUS=0
         NCK=N
         NASA=1
 216     M1=1
         KAPA=0
 217     L3=N-1
 218     DO 1000 LV=M1,L3
         KP2=LV+1
         DO 280 N3=KP2,N
         DO 270 J=1,MJ
         IF(K(LV,J).EQ.K1)GO TO 221
         IF(K(LV,J).EQ.K2)GO TO 222
         IF(K(LV,J).EQ.K4)GO TO 223
```

```
      IF(K(N3,J).EQ.K1)GO TO 223
      IF(K(N3,J).EQ.K2)GO TO 225
      GO TO 221
222   IF(K(N3,J).EQ.K1)GO TO 223
      IF(K(N3,J).EQ.K2)GO TO 221
      GO TO 225
221   KA(LV,J)=K(N3,J)
      GO TO 270
223   KA(LV,J)=K(LV,J)
      GO TO 270
225   KA(LV,J)=K4
270   CONTINUE
      N1=0
      N2=0
      DO 276 J=1,MJ
      IF(KA(LV,J).EQ.K(LV,J))GO TO 272
      IF(KA(LV,J).EQ.K(N3,J))GO TO 275
      GO TO 276
272   IF(KA(LV,J).EQ.K(N3,J))GO TO  273
      GO TO 274
273   N1=N1+1
      N2=N2+1
      GO TO 276
274   N1=N1+1
      GO TO 276
275   N2=N2+1
276   CONTINUE
      IF(N1.EQ.MJ)GO TO 278
      IF(N2.EQ.MJ)GO TO 284
      GO TO 280
278   IF(N2.EQ.MJ)GO TO 283
      DO 310 I=LV,L3
      DO 310 J=1,MJ
310   K(I,J)=K(I+1,J)
      KAPA=KAPA+1
      IF(LV.GE.L3)GO TO 282
      M1=LV
      N=N-1
      L2=N-1
      GO TO 217
282   N=N-1
      L2=N-1
      GO TO 1000
283   NUS=89
284   IF(N3.GE.N)GO TO 287
      DO 286 I=N3,L3
      DO 286 J=1,MJ
286   K(I,J)=K(I+1,J)
287   IF(LV.GE.L3)GO TO 282
      N=N-1
      L3=N-1
      L2=N-1
      KAPA=KAPA+1
      M1=LV
      GO TO 218
280   CONTINUE
1000  CONTINUE
      IF(KAPA.EQ.0)GO TO 292
      IF(NUS.EQ.89)GO TO 293
      GO TO 295
292   GO TO (296,293),NASA
293   IF(NK2.GE.N)GO TO 400
       GO TO 294
294   KAP=NK2+1
      GO TO 297
295   KAP=NK2
      GO TO 297
296   M2=1
      NASA=2
      L2=N-1
      DO 450 MAN=M2,L2
      KAP=MAN+1
297   DO 400 NK2=KAP,N
      NONCE=0
      DO 307 J=1,MJ
      IF(K(MAN,J).EQ.K1)GO TO 304
      IF(K(MAN,J).EQ.K2)GO TO 302
      IF(K(MAN,J).EQ.K4)GO TO 305
      IF(K(NK2,J).EQ.K1)GO TO 305
      IF(K(NK2,J).EQ.K2)GO TO 306
```

```
              GO TO 304
      302     IF(K(NK2,J).EQ.K1)GO TO 305
              IF(K(NK2,J).EQ.K2)GO TO 304
              GO TO 306
      304     KA(MAN,J)=K(NK2.J)
              GO TO 307
      305     KA(MAN,J)=K(MAN,J)
              GO TO 307
      306     KA(MAN,J)=K4
              NACHO=J
              NONCE=NONCE+1
      307     CONTINUE
              IF(NONCE.EQ.1)GO TO 308
              GO TO 400
      308     DO 309J2=1,MJ
      309     K(N+1.J2)=KA(MAN,J2)
              K(N+1,NACHO)=K1
              N=N+1
              L2=N-1
              GO TO 216
      400     CONTINUE
      450     CONTINUE
C
C   IMPRIME TABLA DE IMPLICANTES PRIMOS
C
              WRITE(3,106)
              DO 311L0=1,N
      311     WRITE(3,103)(K(L0,J0),J0=1,MJ)
C
C   FORMA LA MATRIZ DE CUBRIMIENTO
C
              DO 327 M=1,N
              DO 326 NO=1,NUD
              NSTER=0
              DO 323 J=1,MJ
              IF(K(M.J).EQ.K1)GO TO 318
              IF(K(M,J).EQ.K2)GO TO 316
              IF(K(M.J).EQ.K4)GO TO 319
              IF(IB(NO,J).EQ.K1)GO TO 319
              IF(IB(NO,J).EQ.K2)GO TO 320
              GO TO 318
      316     IF(IB(NO,J).EQ.K1)GO TO 319
              IF(IB(NO,J).EQ.K2)GO TO 318
              GO TO 320
      318     KA(M,J)=IB(NO,J)
              GO TO 321
      319     KA(M,J)=K(M,J)
              GO TO 321
      320     KA(M,J)=K4
      321     IF(KA(M,J).EQ.IB(NO.J))GO TO 322
              GO TO 323
      322     NSTER=NSTER+1
      323     CONTINUE
              IF(NSTER.EQ.MJ)GO TO 324
              GO TO 325
      324     KB(M,NO)=1
              GO TO 326
      325     KB(M,NO)=0
      326     CONTINUE
      327     CONTINUE
              WRITE(3,105)
C
C SELECCIONA LOS IMPLICANTES PRIMOS ESCENCIALES
C
              DO 331 KNM=1,N
      331     IK(KNM)=0
              DO 335 NO=1,NUD
              KSUCO=0
              DO 333 M=1,N
              IF(KB(M,NO).EQ.1)GO TO 332
              GO TO 333
      332     NUMER=M
              KSUCO=KSUCO+1
      333     CONTINUE
              IF(KSUCO.EQ.1)GO TO 334
              GO TO 335
      334     IK(NUMER)=1
      335     CONTINUE
              DO 336 KOLA=1,N
      336     KM(KOLA)=9999
```

```
      KARA=1
      DO 338 KNM=1,N
      IF(IK(KNM).EQ.1)GO TO 337
      GO TO 338
C
C   IMPRIME TABLA DE IMPLICANTES PRIMOS ESCENCIALES
C
337   WRITE(3,103)(K(KNM,J),J=1,MJ)
      KM(KARA)=KNM
      KARA=KARA+1
      NIMPR=KARA
      NUIPR=KARA
338   CONTINUE
      JN=0
      DO 343 M=1,N
      IF(IK(M).EQ.1)GO TO 341
      DO 340 J=1,MJ
340   IB(KARA,J)=K(M,J)
      KARA=KARA+1
      GO TO 343
341   JN=JN+1
      DO 342 J=1,MJ
342   IE(JN,J)=K(M,J)
343   CONTINUE
      JR=0
      DO 348 M=1,N
      IF(IK(M).EQ.1)GO TO 346
      DO 345 J=1,NUD
345   K(NIMPR,J)=KB(M,J)
      NIMPR=NIMPR+1
      GO TO 348
346   JR=JR+1
      DO 347 J=1,NUD
347   K(JR,J)=KB(M,J)
348   CONTINUE
C
C   SELECCIONA UN GRUPO DE IMPLICANTES Y PRUBA SI CUBREN LA
C
      NPRIM=NUIPR
      NFUN=0
      DO 349 NO=1,NUD
349   NK(NO)=0
      KOMTA=1
      MA=NUIPR-1
      DO 351 M=1,MA
      DO 350 NO=1,NUD
350   NK(NO)=NK(NO)+K(M,NO)
351   CONTINUE
352   DO 353 NO=1,NUD
      IF(NK(NO).LE.0)GO TO 357
353   CONTINUE
      NFUN=NFUN+1
C
C   IMPRIME UNA FUNCION IMPLIMENTADORA
C
      WRITE(3,107)NFUN
      KROCA=NUIPR-1
      DO 354 NMA=1,KROCA
354   WRITE(3,103)(IB(NMA,J),J=1,MJ)
      DO 355 J=1,MJ
      IF(IB(KROCA,J).EQ.IB(MA,J))GO TO 355
      GO TO 356
355   CONTINUE
      GO TO 357
356   WRITE(3,103)(IB(MA,J),J=1,MJ)
357   IF(NUIPR-1.GE.N)GO TO 7000
      IF(KOMTA.EQ.1)GO TO 361
      DO 360 NO=1,NUD
360   NK(NO)=NK(NO)-K(MA,NO)
361   IF(MA.GE.N)GO TO 370
368   MA=MA+1
      KOMTA=3
      DO 369 NO=1,NUD
369   NK(NO)=NK(NO)+K(MA,NO)
      GO TO 352
370   NUIPR=NUIPR+1
      KOMTA=1
      MA=NUIPR-1
      DO 371 M=1,MA
      DO 371 NO=1,NUD
```

```
0281      371    NK(NO)=NK(NO)+K(M,NO)
0282             IF(MA.GE.N)GO TO 7000
0283             GO TO 368
0284      7000   CONTINUE
0285             END
```

DOS FORTRAN IV 360N-FO-479 3-8 MAINPGM DATE 21/10/76

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SY
K1	444	K2	448	K3	44C	K4
NTR	458	NI	45C	ND	460	MJ
N	46C	M	470	J	474	NA
NUD	480	ISA	484	KAS	488	NU
NASA	494	M1	498	KAPA	49C	L3
KP2	4A8	N3	4AC	N1	4B0	N2
L2	4BC	NK2	4C0	KAP	4C4	M2
NONCE	4D0	NACHO	4D4	J2	4D8	LO
NO	4E4	NSTER	4E8	KNM	4EC	KS
KOLA	4F8	KARA	4FC	NIMPR	500	NU
JR	50C	NPRIM	510	NFUN	514	KO
KROCA	520	NMA	524			

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SY
KA	528	K	B68	KB	1338	IK
KM	2080	NK	20F8			

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SY
IBCOM#	2170					

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SY
100	2174	101	217A	102	2180	
106	21E0	107	2224	108	2264	

EJEMPLOS

Ejemplo N° 1

Se trata de diseñar un conmutador telefónico hipotético, capaz de comunicar entre si a los teléfonos denominados N° 1 y N° 2, que cumpla con las siguientes condiciones:

$$S_{12} = \begin{cases} 1 \text{ si el conmutador conecta N° 1 con N° 2} \\ 0 \text{ si no conecta N° 1 con N° 2} \end{cases}$$

$$Ph_1 = \begin{cases} 1 \text{ si el auricular N° 1 está levantado} \\ 0 \text{ si el auricular N° 1 está colgado} \end{cases}$$

$$Ph_2 = \begin{cases} 1 \text{ si el auricular N° 2 está levantado} \\ 0 \text{ si el auricular N° 2 está colgado} \end{cases}$$

$$H_1 = \begin{cases} 1 \text{ si el teléfono N° 1 está hablando} \\ 0 \text{ si el teléfono N° 1 no está hablando} \end{cases}$$

$$H2 = \begin{cases} 1 \text{ si el teléfono N° 2 está hablando} \\ 0 \text{ si el teléfono N° 2 no está hablando} \end{cases}$$

$$M_1 = \begin{cases} 1 \text{ si N° 1 es llamado} \\ 0 \text{ si N° 1 no es llamado} \end{cases}$$

$$M_2 = \begin{cases} 1 \text{ si N° 2 es llamado} \\ 0 \text{ si N° 2 no es llamado} \end{cases}$$

$$S(m-1)_{12} \begin{cases} 1 \text{ si N° 1 y N° 2 están conectados previamente} \\ 0 \text{ si N° 1 y N° 2 no están conectados previamente} \end{cases}$$

Vemos entonces que S_{12} que es la salida, depende de siete variables, incluyendo su propio estado anterior.

Con estas condiciones podemos construir una tabla de verdad. De los 128 términos posibles, sólo 3 nos dan una salida igual a 1, y corresponden a las siguientes situaciones:

1) Los teléfonos N° 1 y N° 2 no han sido previamente conectados entre si.

$$s(m-1)_{12} = 0$$

Los auriculares N/1 y N° 2 estan levantados

$$Ph_1 = Ph_2 = 1$$

Por ninguno de los teléfonos se está hablando

$H_1 = H_2 = 0$

Teléfono N° 1 no es llamado

$M_1 = 0$,

Teléfono N° 2 es llamado

$M_2 = 1$

2) Los teléfonos N° 1 y N° 2 no han sido previamente conectados entre si.

$S(m-1)_{12} = 0$

Los auriculares N° 1 y N° 2 estan levantados

$Ph_1 = Ph_2 = 1$

Por ninguno de los teléfonos se está hablando

$H_1 = H_2 = 0$

Teléfono N° 1 es llamado

$M_1 = 1$

Teléfono N° 2 no es llamado

$M_2 = 0$

3) La coneçción entre N° 1 y N° 2 es mantenida

$S(m-1)_{12} = 1$

Auriculares Nº 1 y Nº 2 estan levantados

$$Ph_1 = Ph_2 = 1$$

Por los dos teléfonos se está hablando

$$H_1 = H_2 = 1$$

Ningún teléfono es llamado

$$M_1 = M_2 = 0$$

Existen además 56 condiciones indefinidas debido a situaciones que no pueden darse, v.g. no pueden existir dos teléfonos llamados simultaneamente entre si, etc.

Este ejemplo fue corrido en el programa correspondiente al Método de Quine-MacCluskey.

02.56,TOTAL COMPILATION TIME,00.01.08

*FUNCION A MINIMIZAR POR EL METODO DE QUINE-MACLUSKEY

1	0	1	1	0	0	0	1
2	0	1	1	0	0	1	0
3	1	1	1	1	1	0	0

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

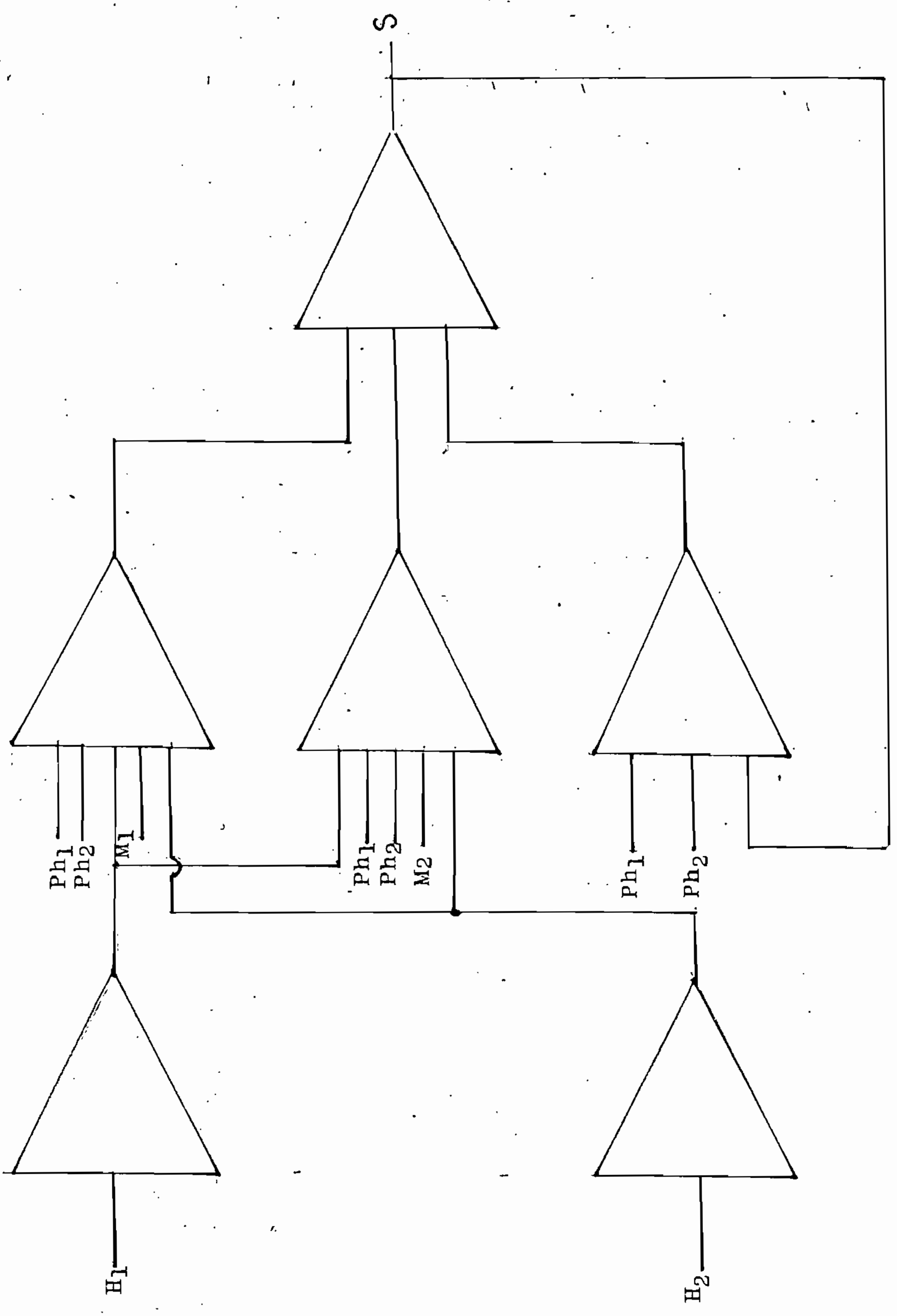
4	1	0	0	0	0	0	0
5	1	0	0	0	0	0	1
6	1	0	0	0	0	1	0
7	1	0	0	0	0	1	1
8	1	0	0	0	1	0	0
9	1	0	0	0	1	0	1
10	1	0	0	0	1	1	0
11	1	0	0	0	1	1	1
12	1	0	0	1	0	0	0
13	1	0	0	1	0	0	1
14	1	0	0	1	0	1	0
15	1	0	0	1	0	1	1
16	1	0	1	0	0	0	0
17	1	0	1	0	0	0	1
18	1	0	1	0	0	1	0
19	1	0	1	0	0	1	1
20	1	0	1	0	1	0	0
21	1	0	1	0	1	0	1
22	1	0	1	0	1	1	0
23	1	0	1	0	1	1	1
24	1	0	1	1	0	0	0
25	1	0	1	1	0	0	1
26	1	0	1	1	0	1	0
27	1	0	1	1	0	1	1
28	1	0	1	1	1	0	1
29	1	0	1	1	1	1	1
30	1	1	0	0	0	0	0
31	1	1	0	0	0	0	1
32	1	1	0	0	0	1	0
33	1	1	0	0	0	1	1
34	1	1	0	0	1	0	0
35	1	1	0	0	1	0	1
36	1	1	0	0	1	1	0
37	1	1	0	0	1	1	1
38	1	1	0	1	0	0	0
39	1	1	0	1	0	0	1
40	1	1	0	1	0	1	0
41	1	1	0	1	0	1	1
42	1	1	0	1	1	1	0
43	1	1	0	1	1	1	1
44	1	1	1	0	0	0	0
45	1	1	1	0	0	0	1
46	1	1	1	0	0	1	0
47	1	1	1	0	0	1	1
48	1	1	1	0	1	0	0
49	1	1	1	0	1	0	1
50	1	1	1	0	1	1	0
51	1	1	1	0	1	1	1
52	1	1	1	1	0	0	0
53	1	1	1	1	0	0	1
54	1	1	1	1	0	1	0
55	1	1	1	1	0	1	1
56	1	1	1	1	1	0	1
57	1	1	1	1	1	1	0
58	1	1	1	1	1	1	1
59	0	1	1	0	0	1	1

Los resultados nos indican que no existen implicantes primos elegibles y que por lo tanto, la función está cubierta unicamente por los implicantes primos esenciales. La salida S_{12} viene expresada en lógica combinada como:

$$S_{12} = Ph_1Ph_2\overline{H}_1\overline{H}_2M_1 + Ph_1Ph_2\overline{H}_1\overline{H}_2M_2 + S(m-1)_{12}Ph_1Ph_2$$

y en lógica NAND será:

$$S_{12} = \overline{\overline{Ph_1Ph_2\overline{H}_1\overline{H}_2M_1} \cdot \overline{Ph_1Ph_2\overline{H}_1\overline{H}_2M_2} \cdot \overline{S(m-1)_{12}Ph_1Ph_2}}$$



Realización en lógica NAND del Ejemplo Nº 1

EJEMPLO Nº 2.

La Fig.   representa un Display de 7 elementos de un Decodificador Digital-Analógico. En el van a representarse los números 1,2,3,4,5,6,7,8,9, y 0 y las letras "E" (error) "F"(lleno) y el simbolo "-" (negativo).

	Representación digital				Representación Analógica
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	-
11	1	0	1	1	E
12	1	1	0	0	F
13	1	1	0	1	i (indefinido)
14	1	1	1	0	i (indefinido)
16	1	1	1	1	i (indefinido)

De la tabla anterior en la que constan las relaciones dígito-analógicas del Decodificador en mención, obtenemos la Tabla de Verdad correspondiente a los siete elementos.

Entradas				Salidas						
x_1	x_2	x_3	x_4	z_a	z_b	z_c	z_d	z_e	z_f	z_g
0	0	0	0	1	0	1	1	1	1	1
0	0	0	1	0	0	0	0	1	0	1
0	0	1	0	1	1	1	0	1	1	0
0	0	1	1	1	1	1	0	1	0	1
0	1	0	0	0	1	0	1	1	0	1
0	1	0	1	1	1	1	1	0	0	1
0	1	1	0	0	1	1	1	0	1	1
0	1	1	1	1	0	0	0	1	0	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	0	1
1	0	1	0	0	1	0	0	0	0	0
1	0	1	1	1	1	1	1	0	1	0
1	1	0	0	1	1	0	1	0	1	0
1	1	0	1	indefinido						
1	1	1	0	indefinido						
1	1	1	1	indefinido						

Podemos ahora obtener las funciones implementadoras de las salidas, para ser procesadas por el computador

10.56.03. TOTAL COMPILATION TIME 00.00.58

$\alpha =$ FUNCION A MINIMIZAR POR EL METODO DE QUINE-MACLUSKEY

1	0	0	0	0
2	0	0	1	0
3	0	0	1	1
4	0	1	0	1
5	0	1	1	1
6	1	0	0	0
7	1	0	0	1
8	1	0	1	1
9	1	1	0	0

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

10	1	1	0	1
11	1	1	1	0
12	1	1	1	1

TABLA DE IMPLICANTES PRIMOS

1	0	0	*	0
2	*	0	0	0
3	0	0	1	*
4	*	*	1	1
5	*	1	*	1
6	1	*	0	*
7	1	*	*	1
8	1	1	*	*

TABLA DE IMPLICANTES PRIMOS ESENCIALES

1	*	1	*	1

TABLA DE IMPLICANTES PRIMOS ELEGIBLES

1	0	0	*	0
2	*	0	0	0
3	0	0	1	*
4	*	*	1	1
5	1	*	0	*
6	1	*	*	1
7	1	1	*	*

TABLA REDUCIDA DE SELECCION

1	*	*	-	-	-	-	-
2	*	-	-	*	-	-	-
3	-	*	*	-	-	-	-
4	-	-	*	-	-	*	-
5	-	-	-	*	*	-	*
6	-	-	-	-	*	*	-
7	-	-	-	-	-	-	*

B = FUNCION A MINIMIZAR POR EL METODO DE QUINE-MACLUSKEY

1	0	0	1	0
2	0	0	1	1
3	0	1	0	0
4	0	1	0	1
5	0	1	1	0
6	1	0	0	0
7	1	0	0	1
8	1	0	1	0
9	1	0	1	1
10	1	1	0	0

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

11	1	1	0	1
12	1	1	1	0
13	1	1	1	1

TABLA DE IMPLICANTES PRIMOS

1	*	0	1	*
2	*	*	1	0
3	*	1	0	*
4	*	1	*	0
5	1	*	*	*

TABLA DE IMPLICANTES PRIMOS ESENCIALES

1	*	0	1	*
2	*	1	0	*
3	1	*	*	*

TABLA DE IMPLICANTES PRIMOS ELEGIBLES

1	*	*	1	0
2	*	1	*	0

TABLA PEDUCIDA DE SELECCION

1	*
2	*

C = FUNCION A MINIMIZAR POR EL METODO DE QUINE-MACLUSKEY

1	0	0	0	0
2	0	0	1	0
3	0	0	1	1
4	0	1	0	1
5	0	1	1	0
6	1	0	0	0
7	1	0	1	1

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

8	1	1	0	1
9	1	1	1	0
10	1	1	1	1

TABLA DE IMPLICANTES PRIMOS

1	0	0	*	0
2	*	0	0	0
3	0	0	1	*
4	0	*	1	0
5	*	0	1	1
6	*	1	0	1
7	*	1	1	0
8	1	*	1	1
9	1	1	*	1
10	1	1	1	*

TABLA DE IMPLICANTES PRIMOS ESENCIALES

1	*	1	0	1
2	*	0	0	0

TABLA DE IMPLICANTES PRIMOS ELEGIBLES

1	0	0	*	0
2	0	0	1	*
3	0	*	1	0
4	*	0	1	1
5	*	1	1	0
6	1	*	1	1

1	*			
2	*	*	—	—
3	*	—	*	—
4	—	*	—	*
5			*	
6	—	—	—	*

$d =$ FUNCION A MINIMIZAR POR EL METODO DE QUINE-MACLUSKEY

1	0	0	0	0
2	0	1	0	0
3	0	1	0	1
4	0	1	1	0
5	1	0	0	0
6	1	0	0	1
7	1	0	1	1
8	1	1	0	0

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

9	1	1	0	1
10	1	1	1	0
11	1	1	1	1

TABLA DE IMPLICANTES PRIMOS

1	*	*	0	0
2	*	1	0	*
3	*	1	*	0
4	1	*	0	*
5	1	*	*	1
6	1	1	*	*

TABLA DE IMPLICANTES PRIMOS ESENCIALES

1	*	*	0	0
2	*	1	0	*
3	*	1	*	0
4	1	*	*	1

$e =$ FUNCION A MINIMIZAR POR EL METODO DE QUINE-MACLUSKEY

1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	1	1
7	1	0	0	0
8	1	0	0	1

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

9	1	1	0	1
10	1	1	1	0
11	1	1	1	1

TABLA DE IMPLICANTES PRIMOS

1	0	0	*	*
2	0	*	0	0
3	*	0	0	*
4	0	*	1	1
5	*	1	1	1
6	1	*	0	1
7	1	1	*	1
8	1	1	1	*

TABLA DE IMPLICANTES PRIMOS ESENCIALES

1	0	0	*	*
2	0	*	0	0
3	*	0	0	*

TABLA DE IMPLICANTES PRIMOS ELEGIBLES

1	0	*	1	1
2	*	1	1	1

TABLA REDUCIDA DE SELECCION

1	*
2	*

$f =$

FUNCION A MINIMIZAR POR EL METODO DE QUINE-MACLUSKEY

1	0	0	0	0
2	0	0	1	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

7	1	1	0	1

TABLA DE IMPLICANTES PRIMOS

1	0	0	*	0
2	*	0	0	0
3	0	*	1	0
4	*	1	1	0
5	1	*	0	0
6	1	*	1	1
7	1	1	*	*

TABLA DE IMPLICANTES PRIMOS ESENCIALES

1	1	*	1	1

TABLA DE IMPLICANTES PRIMOS ELEGIBLES

1	0	0	*	0
2	*	0	0	0
3	0	*	1	0
4	*	1	1	0
5	1	*	0	0
6	1	1	*	*

TABLA REDUCIDA DE SELECCION

1	*	*	—	—	—
2	*	—	—	*	—
3	—	*	*	—	—
4	—	—	*	—	—
5	—	—	—	*	*
6	—	—	—	—	*

$f =$ FUNCION A MINIMIZAR POR EL METODO DE QUINE-MACLUSKEY

1	0	0	0	0
2	0	0	0	1
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

10	1	1	0	1
11	1	1	1	0
12	1	1	1	1

TABLA DE IMPLICANTES PRIMOS

1	0	*	0	*
2	*	0	0	*
3	0	*	*	1
4	*	*	0	1
5	0	1	*	*
6	*	1	*	1
7	*	1	1	*

TABLA DE IMPLICANTES PRIMOS ESENCIALES

1	0	*	*	1
2	*	0	0	*

TABLA DE IMPLICANTES PRIMOS ELEGIBLES

1	0	*	0	*
2	0	1	*	*
3	*	1	1	*

TABLA REDUCIDA DE SELECCION

1	*	—
2	*	*
3	—	*

EPN MSJ5 TIEMPO DE UCP UTILIZADO POR EL PROGRAMA LUCIO ------> 55,6

Nos resta interpretar las Tablas de Selección generadas por el Computador y determinar las mínimas coberturas para las siete funciones del ejemplo.

A continuación se da las mínimas expresiones correspondientes a las funciones representativas del Decodificador.

$$z_a = \overline{\overline{x_2 x_4} \cdot \overline{\bar{x}_1 \bar{x}_2 \bar{x}_4} \cdot \overline{x_3 x_4} \cdot \overline{x_1 \bar{x}_3}}$$

$$z_b = \overline{\overline{\bar{x}_2 x_3} \cdot \overline{x_2 \bar{x}_3} \cdot \overline{x_1} \cdot \overline{x_3 \bar{x}_4}}$$

$$z_c = \overline{\overline{x_2 \bar{x}_3 x_4} \cdot \overline{\bar{x}_2 \bar{x}_3 \bar{x}_4} \cdot \overline{\bar{x}_1 x_3 \bar{x}_4} \cdot \overline{\bar{x}_2 x_3 x_4}}$$

$$z_d = \overline{\overline{\bar{x}_3 \bar{x}_4} \cdot \overline{x_2 \bar{x}_3} \cdot \overline{x_2 \bar{x}_4} \cdot \overline{x_1 x_4}}$$

$$z_e = \overline{\overline{\bar{x}_1 \bar{x}_2} \cdot \overline{\bar{x}_1 \bar{x}_3 \bar{x}_4} \cdot \overline{\bar{x}_2 \bar{x}_3} \cdot \overline{x_2 x_3 x_4}}$$

$$z_f = \overline{\overline{x_1 x_3 x_4} \cdot \overline{\bar{x}_2 \bar{x}_3 \bar{x}_4} \cdot \overline{\bar{x}_1 x_3 \bar{x}_4} \cdot \overline{x_1 x_2}}$$

$$z_g = \overline{\overline{\bar{x}_1 x_4} \cdot \overline{\bar{x}_2 \bar{x}_3} \cdot \overline{\bar{x}_1 x_2}}$$

<u>EJEMPLO N° 3</u>

Este ejemplo tiene por objeto utilizar el programa correspondiente al método de Mott y comparar los resultados obtenidos mediante el de Quine-McCluskey. Para ello usamos cuatro de las funciones del ejmploN° 2 y son aquellas que corresponden a: z_a, z_b, z_c y z_d .

El programa de Mott nos da los siguientes resultados para estas cuatro funciones:

$$z_a = \overline{\overline{x_1x_2\bar{x}_3\bar{x}_4} \cdot \overline{\bar{x}_2\bar{x}_3\bar{x}_4} \cdot \overline{x_1x_4} \cdot \overline{x_2x_3} \cdot \overline{\bar{x}_1\bar{x}_2x_3}}$$

$$z_b = \overline{\overline{x_1\bar{x}_2\bar{x}_3\bar{x}_4} \cdot \overline{\bar{x}_1\bar{x}_2x_3} \cdot \overline{x_1x_4} \cdot \overline{\bar{x}_1x_2\bar{x}_3} \cdot \overline{x_2\bar{x}_3\bar{x}_4} \cdot \overline{x_3\bar{x}_4}}$$

$$z_c = \overline{\overline{\bar{x}_1x_2\bar{x}_3x_4} \cdot \overline{\bar{x}_2\bar{x}_3\bar{x}_4} \cdot \overline{x_1x_2\bar{x}_3x_4} \cdot \overline{\bar{x}_1\bar{x}_2\bar{x}_4} \cdot \overline{\bar{x}_2x_3x_4}}\ \cdot$$

$$\overline{\overline{x_2x_3\bar{x}_4}}$$

$$z_d = \overline{\overline{x_1\bar{x}_2\bar{x}_3\bar{x}_4} \cdot \overline{\bar{x}_1\bar{x}_3\bar{x}_4} \cdot \overline{\bar{x}_1x_2\bar{x}_4} \cdot \overline{x_2\bar{x}_3} \cdot \overline{x_1x_4}}$$

Los resultados anteriores provienen de la menor función implementadora de las generadas para cada caso.

Si comparamos los resultados obtenidos por los dos programas, salta a la vista que el programa correspondiente al Método Tabular ha generado en todos los cuatro casos funciones implementadoras menores, lo que nos lleva a pensar que este programa, cuando se parte de la función expandida da mejores resultados.

FUNCION BOOLEANA A MINIMIZAR
ESTA REPRESENTADA POR LA SUMA DE LOS SIGUIENTES PRODUCTOS

a

01	01	01	01
01	01	10	01
01	01	10	10
01	10	01	10
01	10	10	10
10	01	01	01
10	01	01	10
10	01	10	10
10	10	01	01

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

10	10	01	10
10	10	10	01
10	10	10	10

TABLA DE IMPLICANTES PRIMOS METODO DE MOTT

10	10	01	01
01	01	00	01
00	01	01	01
10	10	10	00
01	01	10	00
00	10	00	10
10	00	00	10
00	00	10	10

TABLA DE IMPLICANTES PRIMOS ESENCIALES

10	10	01	01
00	01	01	01
00	10	00	10
10	00	00	10

FUNCION IMPLEMENTADORA NUMERO 1

10	10	01	01
00	01	01	01
00	10	00	10
10	00	00	10
01	01	10	00

FUNCION IMPLEMENTAOORA NUMERO 2

10	10	01	01
00	01	01	01
00	10	00	10
10	00	00	10
01	01	00	01
01	01	10	00

FUNCION IMPLEMENTADORA NUMERO 3

10	10	01	01
00	01	01	01
00	10	00	10
10	00	00	10
01	01	00	01
00	00	10	10

FUNCION IMPLEMENTADORA NUMERO 4

10	10	01	01
00	01	01	01
00	10	00	10
10	00	00	10
01	01	00	01
10	10	10	00
01	01	10	00

FUNCION IMPLEMENTADORA NUMERO 5

10	10	01	01
00	01	01	01
00	10	00	10
10	00	00	10
01	01	00	01
10	10	10	00
00	00	10	10

FUNCION IMPLEMENTADORA NUMERO 6

10	10	01	01
00	01	01	01
00	10	00	10
10	00	00	10
01	01	00	01
10	10	10	00
01	01	10	00
00	00	10	10

FUNCION BOOLEANA A MINIMIZAR

b ESTA REPRESENTADA POR LA SUMA DE LOS SIGUIENTES PRODUCTOS

01	01	10	01
01	01	10	10
01	10	01	01
01	10	01	10
01	10	10	01
10	01	01	01
10	01	01	10
10	01	10	01
10	01	10	10
10	10	01	01

. LAS SIGUIENTES SON CONDICIONES INDEFINIDAS
==

```
10   10   01   10
10   10   10   01
10   10   10   10
```

TABLA DE IMPLICANTES PRIMOS   METODO DE MOTT
****

```
10   01   01   01
01   01   10   00
01   10   01   00
00   10   01   01
01   10   00   01
00   10   01   10
00   00   10   01
10   00   00   10
10   00   10   00
10   10   00   00
```

TABLA DE IMPLICANTES PRIMOS ESENCIALES
****

```
10   01   01   01
01   01   10   00
10   00   00   10
```

==

FUNCION IMPLEMENTADORA NUMERO 1
===================================

```
10   01   01   01
01   01   10   00
10   00   00   10
01   10   01   00
00   10   01   01
00   00   10   01
```

==

FUNCION IMPLEMENTADORA NUMERO 2
===================================

```
10   01   01   01
01   01   10   00
10   00   00   10
01   10   01   00
00   10   01   01
01   10   00   01
00   00   10   01
```

==

FUNCION IMPLEMENTADORA NUMERO 3
===================================

```
10   01   01   01
01   01   10   00
10   00   00   10
01   10   01   00
00   10   01   01
01   10   00   01
10   00   10   00
```

FUNCION IMPLEMENTADORA NUMERO 4

10	01	01	01
01	01	10	00
10	00	00	10
01	10	01	00
00	10	01	01
01	10	00	01
00	10	01	10
00	00	10	01

FUNCION IMPLEMENTADORA NUMERO 5

10	01	01	01
01	01	10	00
10	00	00	10
01	10	01	00
00	10	01	01
01	10	00	01
00	10	01	10
10	00	10	00

FUNCION IMPLEMENTADORA NUMERO 6

10	01	01	01
01	01	10	00
10	00	00	10
01	10	01	00
00	10	01	01
01	10	00	01
00	10	01	10
00	00	10	01
10	00	10	00

FUNCION IMPLEMENTADORA NUMERO 7

10	01	01	01
01	01	10	00
10	00	00	10
01	10	01	00
00	10	01	01
01	10	00	01
00	10	01	10
00	00	10	01
10	10	00	00

FUNCION IMPLEMENTADORA NUMERO 8.

10	01	01	01
01	01	10	00
10	00	00	10
01	10	01	00
00	10	01	01
01	10	00	01
00	10	01	10
00	00	10	01
10	00	10	00
10	10	00	00

FUNCION BOOLEANA A MINIMIZAR
ESTA REPRESENTADA POR LA SUMA DE LOS SIGUIENTES PRODUCTOS

01	01	01	01
01	01	10	01
01	01	10	10
01	10	01	10
01	10	10	01
10	01	01	01
10	01	10	10

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

10	10	01	10
10	10	10	01
10	10	10	10

TABLA DE IMPLICANTES PRIMOS METODO DE MOTT

01	10	01	10
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
00	01	01	01
10	00	10	10
01	00	10	01
01	01	10	00
10	10	10	00

TABLA DE IMPLICANTES PRIMOS ESENCIALES

01	10	01	10
00	01	01	01

FUNCION IMPLEMENTADORA NUMERO 1

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01

UNCION IMPLEMENTADORA NUMERO 2

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
01	00	10	01

FUNCION IMPLEMENTADORA NUMERO 3

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
10	00	10	10

FUNCION IMPLEMENTADORA NUMERO 4

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
01	00	10	01

FUNCION IMPLEMENTADORA NUMERO 5

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
01	01	10	00

UNCION IMPLEMENTADORA NUMERO 6

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
10	10	10	00

UNCION IMPLEMENTADORA NUMERO 7

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
10	00	10	10
01	00	10	01

UNCION IMPLEMENTADORA NUMERO 8

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
10	00	10	10
01	01	10	00

UNCION IMPLEMENTADORA NUMERO 9

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
10	00	10	10
10	10	10	00

UNCION IMPLEMENTADORA NUMERO 10

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
10	00	10	10
01	00	10	01
01	01	10	00

FUNCION IMPLEMENTADORA NUMERO 11

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
10	00	10	10
01	00	10	01
10	10	10	00

FUNCION IMPLEMENTADORA NUMERO 12

01	10	01	10
00	01	01	01
10	10	01	10
01	01	00	01
00	01	10	10
00	10	10	01
10	00	10	10
01	00	10	01
01	01	10	00
10	10	10	00

FUNCION BOOLEANA A MINIMIZAR
ESTA REPRESENTADA POR LA SUMA DE LOS SIGUIENTES PRODUCTOS

01	01	01	01
01	10	01	01
01	10	01	10
01	10	10	01
10	01	01	01
10	01	01	10
10	01	10	10
10	10	01	01

LAS SIGUIENTES SON CONDICIONES INDEFINIDAS

10	10	01	10
10	10	10	01
10	10	10	10

TABLA DE IMPLICANTES PRIMOS METODO DE MOTT

10	01	01	01
01	00	01	01
01	10	00	01
00	10	01	00
10	00	00	10
10	10	00	00

TABLA DE IMPLICANTES PRIMOS ESENCIALES

10	01	01	01
01	00	01	01
01	10	00	01
00	10	01	00
10	00	00	10

==

NCION IMPLEMENTADORA NUMERO 1
================================

```
10  01  01  01
01  00  01  01
01  10  00  01
00  10  01  00
10  00  00  10
```

==

NCION IMPLEMENTADORA NUMERO 2
================================

```
10  01  01  01
01  00  01  01
01  10  00  01
00  10  01  00
10  00  00  10
10  10  00  00
```

EPN MSJ5 TIEMPO DE UCP UTILIZADO POR EL PROGRAMA LUCIO ------> 69.

CONCLUSIONES

En el presente trabajo se han estudiado métodos de minimiza - ción de la Función Booleana y métodos de minimización en Lógica NAND. De los primeros se han seleccionado para ser programados a los más representativos, el Método Tabular o de Quine MacCluskey y el Método de Moot basado en el Consensus Iterado.

En la parte de minimización en Lógica NAND se han estudiado dos métodos: el Método de Dietmeyer-Su, de realización suma - mente difícil y engorrosa, que no justifica su programación; el Método de Muroga-Ibaraki, muy interesante y versatil, pero que lastimosamente está basado en una codificación desarrollada en la Universidad de Urbana, cuya consecución no ha sido posible, y que ademá requiere de programación lineal entera, tampoco justifica su programación.

Una transformación, basada en los teoremas de Morgan ha sido expuesta. Su sencillez y simplicidad, más el hecho de no re - querir variación alguna de los programas de simplificación de la función Booleana, demuestran su utilidad para los fines que persigue este trabajo.

De los ejemplos se desprende la eficacia de los programas; se observa también que uno de ellos, el correspondiente al Método Tabular, supera al otro en efectividad. En compensación, el programa correspondiente al Método de Mott requiere menos

BIBLIOGRAFIA

1) Bartee - Lebow - Reed. "THEORY AND DESIGN OF DIGITAL MACHINES". McGraw - Hill - 1.962. Pág. 49 - 69

2) Taylos L. Booth. "DIGITAL NETWORKS AND COMPUTER SYSTEMS". John Wiley and Sons, Inc. - 1.971. Pág. 122 - 145.

3) E.J.M. Van Lantschoot. "A PDP-S PROGRAM FOR MINIMISATION OF LOGICAL FUNCTIONS". Artículo presentado en el International Symposium, Bruselas, Septiembre 17 1969.

4) Andrew A. Perlowski. " A MINIMIZATION OF BOOLEAN FUNCTION USING A DIGITAL COMPUTER". IEEE Computer Society - 1970.

5) D.L. Dietmeyer and Y-H Su. "LOGICAL DESIGN AUTOMATION OF GAN-IN LIMITED NAND NETWORKS". IEEE Computer Society - 1968.

6) Saburo Muroga - Toshihide Ibaraki. "DESIGN OF OPTIMAL SWITCHING NETWORKS BY INTEGER PROGRAMING". IEEE Transactions on Computers, Vol. C-21, Nº 6, Junio 1972.

memoria para su realización; en efecto, son necesarios catorce mil ochocientos bytes para este programa, en comparación con los treinta y seis mil novecientos noventa y cuatro bytes requeridos por el Método de Quine-McCluskey, es decir menos del 50% para resolver problemas de igual embergadura. En el ejemplo N° 1 es necesario una mayor capacidad de memoria debido a su particular magnitud.

En el Ejemplo número uno, el programa del método tabular, utiliza 102.53 segundos para minimizar una función de cincuenta y nueve productos de los cuales cincuenta y seis son condiciones indefinidas, con siete variables. En el Ejemplo No. 2 el mismo programa emplea 55.68 segundos para minimizar siete funciones de cuatro variables.

En el Ejemplo N° 3, el programa correspondiente al Método de Mott utiliza 59.99 segundos en simplificar cuatro funciones de cuatro variables.

Estos tiempos han sido logrados por un computador I.B.M. 370/125 en el cual han sido corridos los programas.

7) Daniel D. McCracken. "PROGRAMACION FORTRAN IV". Limusa Wiley - 1967.

8) IBM,S.A.E. Departamento de Traducciones y Ediciones. "IBM SISTEMA/360 Y SISTEMA/370 LENGUAJE FORTRAN IV" International Business Machines Corporation, USA - 1971.

9) David Gordon Dutra. "COMPUTER REDUCTION OF BOOLEAN EXPRESSIONS". IEEE Computer Society 1967.

10) Pierre Tison. "APLICATION OF CONCENSUS THEORY TO THE MINIMIZATION OF BOOLEAN FUNCTIONS". IEEE Computer Society 1966.