Qüestiió - V. 7, n.º 1 (març 1983) pp. 401-410

SÍNTESIS DE FUNCIONES DE CONMUTACIÓN CON BLOQUES CUALESQUIERA

A. R. VAQUERO

UNIVERSIDAD COMPLUTENSE DE MADRID

En este trabajo se aborda el problema de la síntesis de funciones de conmutación multisalida, a partir de bloques cualesquiera también de salidas múltiples, e independientemente de las especificaciones (universalidad, completitud, etc.) de dichas funciones. La síntesis de una función multisalida se reduce a la descomposición iterativa de la función en dos etapas con una de ellas conocida, problema que se ataca mediante la resolución de un sistema de ecuaciones booleanas. Puesto que la descomposición de funciones no es conmutativa, se tratan por separado los casos en los que la primera o la segunda etapa están prefijadas.

SYNTHESIS OF CONMUTATION FUNCTIONS WITH ANY TYPE BLOCKS

Key words: C.A.D. (Computer Aided Design). Boole's Algebra
Modular synthesis of digital circuits.

1. INTRODUCCION.

La síntesis modular de circuitos de conmutación sirve no sólo para construir sistemas digitales con módulos previamente fabricados, sino también para diseñar los módulos integrados que se han de fabricar. La síntesis de circuitos combinacionales es un problema que ha sido abordado desde diferentes perspectivas.

Cuando no se fijan los bloques para el diseño sino las etapas y las entradas a cada etapa, se emplea el método basado en los trabajos de Ashenhurst /1/, Curtis /4/ y Deschamps /5/. Dicho método ha sido aplicado a la descomposición de funciones simples por Curtis /4/. Para funciones múltiples se ha aplicado por vez primera, según lo que conocemos, en /6/. Este método de descomposición también ayuda, por su potencialidad, a resolver ciertos problemas particulares con bloques fijados, sobre todo si se diseña con módulos lógicos universales /6/.

Cuando los bloques son conocidos, existen procedimientos de síntesis que dependen de la naturaleza de los bloques. Si los módulos son universales, existen métodos apropiados para cada tipo de módulo (NOR, NAND, WOS, multiplexor, ...). Cuando se trata de un conjunto completo de módulos elementales, se intenta crear el método particular deducido del álgebra propia con los operadores adecuados ({"0 exclusiva", Y}, {"0 exclusiva", "equivalencia lógica"}, ...) /9/, /10/.

En este trabajo se recogen los resultados obtenidos hasta hoy en una línea de trabajo más general, que consiste en la síntesis de funciones de salida múltiple a partir de bloques cualesquiera de salida múltiple; es decir, que los métodos están basados en el conocimiento de las funciones que realizan los bloques dados y son independientes de las especificidades (universalidad o no universalidad, completitud o no completitud, etc.) de tales funciones.

La síntesis de circuitos combinacionales monosalida ha sido estudiada /16/. Para circuitos multisalida a partir de bloques monosalida ha sido estudiada en /3/, aplicando los métodos de resolución de sistemas de ecuaciones booleanas /15/, /11/, /2/ y /12/.

Nosotros hemos abordado el problema de sintetizar una función múltiple a partir de blo--

- A. Vaquero - Dpto. de Informática y Automática de la Universidad Complutense de Madrid Ciudad Universitaria - Madrid

- Article rebut el Gener del 1982.

ples para obtener la función f. En el caso de que la primera etapa se forme con bloques conocidos, se puede, de la misma manera, reducir el problema general a la búsqueda de una función multisalida que sea solución de la segunda etapa, siendo la primera un bloque fijo del catálogo.

3. DESCOMPOSICION DE UNA FUNCION MULTIPLE EN DOS ETAPAS CON UNA DE ELLAS FIJA.

3.1. PLANTEAMIENTO DEL PROBLEMA

Dado un bloque G de p entradas y n salidas, con ecuaciones características: (Fig. 5)

$$s_1 = G_1(x_1, \ldots, x_p)$$
$$\vdots \quad \vdots$$
$$s_n = G_n(x_1, \ldots, x_p)$$

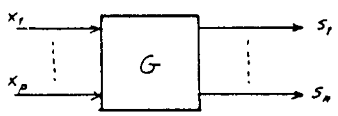

Fig. 5: Bloque G

y un bloque F, de q entradas y n salidas y ecuaciones características: (Fig. 6)

$$r_1 = F_1(a_1, a_2, \ldots, a_q)$$
$$r_2 = F_2(a_1, \ldots, a_q)$$
$$\vdots \quad \vdots$$
$$r_n = F_n(a_1, \ldots, a_q)$$

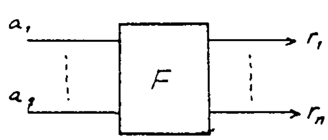

Fig. 6: Bloque F

Se trata de descomponer F en dos etapas, de forma que G sea precisamente la segunda etapa y el circuito combinacional, f, a determinar sea la primera etapa (Fig. 7).

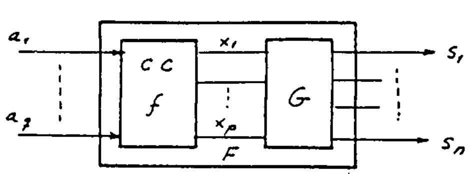

Fig. 7: Descomposición de F en dos etapas, fijada la segunda.

Es decir, debemos hallar unas funciones, $x_j = f_j(a_k)$, que cumplan el sistema:

$$G_i(x_1, \ldots, x_p) = F_i(a_1, \ldots, a_q) \qquad (i=1, \ldots, n)$$

b) Fijamos la primera etapa

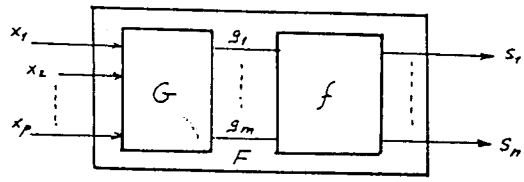

Fig. 8: Descomposición de F en dos etapas fijada la primera.

Tendremos que encontrar las funciones:

$$s_k = f_k(g_e) \qquad \begin{array}{l} k = 1, \ldots, n \\ e = 1, \ldots, m \end{array}$$

que son solución del sistema:

$$F_k(x_j) = S_k(g_e) \qquad j=1, \ldots, p$$

teniendo en cuenta que se debe verificar: $g_e = G(x_j)$.

3.2. RESOLUCION CON LA SEGUNDA ETAPA FIJADA.

En el trabajo de Marín /16/ se plantea y resuelve la síntesis de circuitos combinacionales monosalida con el método de descomposición en dos etapas, manteniendo siempre fija la segunda y utilizando como módulo conocido un tipo de circuito integrado universal monosalida. Para la resolución del problema, se basa en el algoritmo de resolución de sistemas de ecuaciones booleanas de Svoboda /11/. Posteriormente, Marin y Cerny /3/ resuelven el mismo problema pero basándose en el algoritmo de solución reducida del sistema de ecuaciones booleanas dado por Brown /2/. En él abordan también la síntesis de funciones multisalida, pero a partir de bloques fijos monosalida.

3.2.1 El método general

Nosotros hemos estado trabajando en el estudio completo de la síntesis de una función (completa o incompleta) múltiple F, a partir de un módulo conocido G, fijado G como segunda etapa (Fig. 7). A partir de las tablas de verdad de F y G se plantea el sistema de ecuaciones correspondientes:

$$F_i(a_k) = G_i(x_j) \quad o$$

$$\phi_i(x_j, a_k) = \gamma_i(x_j, a_k) \qquad \begin{matrix} i=1 \ldots n \\ j=1 \ldots p \\ k=1 \ldots q \end{matrix} \qquad (1)$$

Se trata, pues, de resolver este sistema de n ecuaciones, o sea, calcular las funciones $x_j = x_j(a_k)$ que satisfagan dicho sistema.

El sistema (1), expresando cada ecuación en la forma "'O exclusiva' de ambos miembros = = 0 ", sumando lógicamente todas las ecuaciones y realizando una expansión según el teorema de Shanon, queda:

$$\sum_{i=0}^{2^p-1} m_i(x_j) \cdot g_i(a_n) = 0 \qquad (2)$$

siendo $m_i(x_j)$ el minterm iésimo de las x's; y $g_i(a_k)$ son 2^p funciones exclusivamente de las a_k, calculadas a partir de γ_i y ϕ_i.

Puede demostrarse, a partir del criterio de Whitehead (15) que la CNS de existencia de - solución es:

$$\prod_{i=0}^{2^p-1} g_i(a_k) = 0 \qquad (3)$$

Se puede comprobar fácilmente que, en el caso de funciones monosalida (3) se cumple --- siempre para una topología de etapas separadas. Obviamente también se cumple, con el -- mismo tipo de topología, para funciones multisalida sintetizadas con bloques monosalida.

Una vez verificada la CNS de existencia de - solución, se hallan las x_j a través de un -- proceso iterativo. En cada etapa hallamos la expresión analítica de una de las x_j:

$$x_j = Q_j + c\, \bar{p}_j$$

donde P_j y Q_j dependen de todas las a_k y todas aquéllas que se han hallado en pasos anteriores;de forma que después de p pasos, obtenemos la solución general del sistema de - ecuaciones propuesto.

3.2.2. Incompatibilidades.

Caso de que la CNS de existencia de solución (3) no se verifique, para una topología de - etapas totalmente separadas (Fig. 7), hemos encontrado un método capaz de determinar el conjunto mínimo de salidas. Si, del bloque F que no pueden ser generadas a partir de G, - sea cual fuere el circuito f. En ese caso -- esas salidas deben ser generadas directamente por f (Fig. 9)

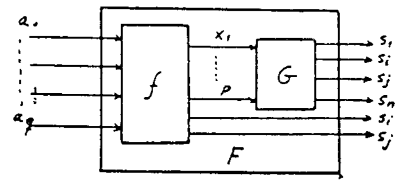

Fig. 9: En caso de incompatibilidad, se generan directamente algunas salidas.

De este modo queda eliminada la incompatibilidad y se sigue aprovechando al máximo la - operatividad del bloque G.

La CNS para la existencia de solución (2) se puede expresar en términos de matrices dis-- criminantes /2/. Dadas las matrices discriminantes de todas y cada una de las salidas, el discriminante de la función completa ---- $F(a_k)$ es la suma lógica de los n discriminantes parciales. El sistema (1) es compatible cuando "el producto lógico de todos los términos de cada una de las columnas del discriminante total es cero". Así pues, cuando el sistema (1) es incompatible, ocurre que el - producto lógico de los términos de al menos una columna del discriminante total es uno, o sea, que todos los términos de alguna o algunas columnas son "1".

Fijándonos en el caso de incompatibilidad, - hemos de señalar las columnas de la matriz discriminante del sistema total cuyos elementos son todos "1" y, para cada una de esas columnas, detectar los discriminantes parciales - responsables de hacer "1" cada elemento de - la columna. Basta con que, en cada una de -- esas columnas, uno de sus elementos sea "0" para hacer compatible el sistema. Para una - columna bastaría con marcar el elemento para el que hay un número menor de discriminantes parciales que tienen "1" en ese lugar, ya -- que las salidas correspondientes a esos discriminantes son el conjunto mínimo buscado. En el caso de más de una columna, todos cu-- yos elementos toman el valor "1", el problema es algo más complejo, aunque similar al - anterior, ya que se trata de encontrar el mínimo número de discriminantes responsables - de hacer "1" al menos un elemento en cada --

una de las citadas columnas.

Una vez descubierto el conjunto mínimo de salidas incompatibles, la solución del nuevo sistema compatible (en el que se han suprimido las ecuaciones correspondientes a salidas incompatibles) nos da la función de conmutación correspondiente a la primera etapa del circuito. Esta función condiciona las entradas a la segunda etapa para obtener correctamente todas las salidas, excepto aquéllas correspondientes al conjunto mínimo incompatible. Estas últimas han de generarse directamente /7/ desde la primera etapa (Fig. 9).

3.2.3. Términos redundantes.

Completando el trabajo, se ha tratado el caso en que F sea incompleta(*). Se trata, sobre todo, de saber cómo influyen los términos redundantes en el método de resolución del sistema, es decir, cómo son los discriminantes parciales y total, y cómo detectar sobre ellos la CNS de existencia de solución o la incompatibilidad del sistema.

Cada salida da lugar a una ecuación:

$$F_i(a_k)=G_i(x_j) \quad '' \quad (i=1,\dots,n), \text{ siendo:} \qquad (4)$$

$$F_i(a_k)=\sum_{k=0}^{2^q-1} e_k^i.m_k(a_1\dots a_q);\ e_k^i \in \{0,1,c\}$$

$$G_i(x_j)=\sum_{j=0}^{2^p-1} d_j^i.m_j(x_1\dots x_p);d_j^i \in \{0,1\}$$

El sistema total, como suma de las n ecuaciones (4), después de realizar la "0 exclusiva" de F_i y G_i en cada una y expandir por Shanon, tomará la forma:

$$\sum_{i=1}^{n}\sum_{1=0}^{2^p-1} h_1^i(a_k).m_1(x_j)=0 \qquad (5)$$

Cada elemento matricial, $D_{1,k}^i$ de un discriminante parcial D^i podrá tomar valores 0,1,c ó $\bar{c}$.

Cada fila de d^i representa los valores h_1^i (1=0, 2^p-1) como funciones de los "minterms" de las variables a_k.

Si en la expresión (5) llamamos $g_1(a_k)\equiv \sum_{i=1}^{n} h_1^i(a_k)$, se puede escribir como:

$$(5')\quad \sum_{1=0}^{2^p-1} g_1(a_k).m_1(x_j) = 0, \text{ de forma que}$$

la CNS de existencia de solución se expresa:

$$\mathcal{M}_{1=0}^{2^p-1}\ g_1(a_k) = 0 \qquad (6)$$

Vamos a interpretar (6) sobre los discriminantes. En el discriminante total, cada columna ha de contener elementos tales que su producto lógico sea "0". En cuanto a las columnas que corresponden a minterms no redundantes de las variables a_k, la condición es la misma que para funciones completas, es decir, al menos un elemento de la columna ha de ser cero. Lo que queremos saber ahora es cómo son las columnas del discriminante total que corresponden a términos redundantes. Para ello veremos antes como son en este caso los discriminantes parciales.

En cada discriminante parcial, D^i existen dos tipos de filas:

a) Una fila, j, de D_i coincidirá con los valores lógicos de F_i, en el caso de que G_i no contenga el minterm de las x's correspondientes a esa fila j.

b) Una fila j de D_i coincidirá con los valores lógicos de $\bar{F}_i$ en el caso de que G_i contenga el minterm de las x's correspondientes a esa fila j.

En cuanto a las columnas, teniendo en cuenta la existencia de redundantes, habrá:

a) Las correspondientes a los minterms de las a's <u>redundantes</u> para F.

b) Las correspondientes a los minterms de las a's que <u>no</u> son redundantes para F.

Sigamos analizando las columnas de tipo a) que son las que a nosotros nos interesan.

En cada discriminante parcial (Fig. 10), estas columnas son idénticas entre sí y compuestas de c's y $\bar{c}$'s.

(*) G es siempre completa por ser un módulo fabricado. No tiene sentido desconocer su repuesta para alguna configuración a su entrada.

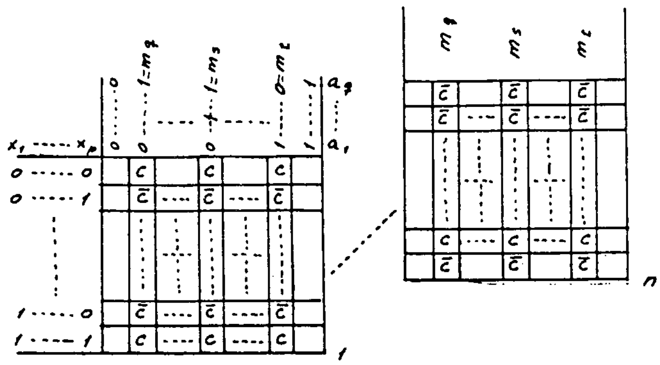

Fig. 10: Matrices discriminantes

Las c's y $\bar{c}$'s de una misma columna, en un mismo D^i, están ligadas por corresponder a la misma F_i y al mismo minterm redundante de las a's.

Por estar ligadas entendemos: que si una c -en la fila q y columna p- significa que para $m_p(a_k)$ el correspondiente $m_q(x_j)$ no debe pertenecer a G_i, entonces $\bar{c}$ deberá significar que el correspondiente $m_q(x_j)$ sí debe pertenecer a G_i o viceversa.

Sin embargo, las c's y $\bar{c}$'s de columnas diferentes en un mismo D^i no están ligadas, ya que pertenecen a minterms redundantes diferentes. Por lo tanto, la asignación de 0's ó 1's a las c's de columnas diferentes no tiene por qué ser la misma.

Las columnas correspondientes a un mismo minterm de las a's en los distintos D^i no están ligadas ni tienen por qué ser iguales entre sí, debido a que pertenecen a diferentes funciones F_i y G_i.

Lógicamente, el discriminante total D (Fig. 11) tendrá el mismo número de columnas redundantes que los D^i y en las mismas posiciones.

Los elementos de cada columna -correspondientes a minterms redundantes- serán sumas lógicas de c's y $\bar{c}$'s. Todas estas columnas serán iguales, puesto que provienen de la suma de los D^i y, como vimos, cada D^i tiene sus columnas de redundantes iguales entre sí.

En cuanto a las columnas correspondientes a términos redundantes, la CNS para la existencia de solución se traduce en que el producto lógico de sus elementos (Fig. 11) ha de ser "0".

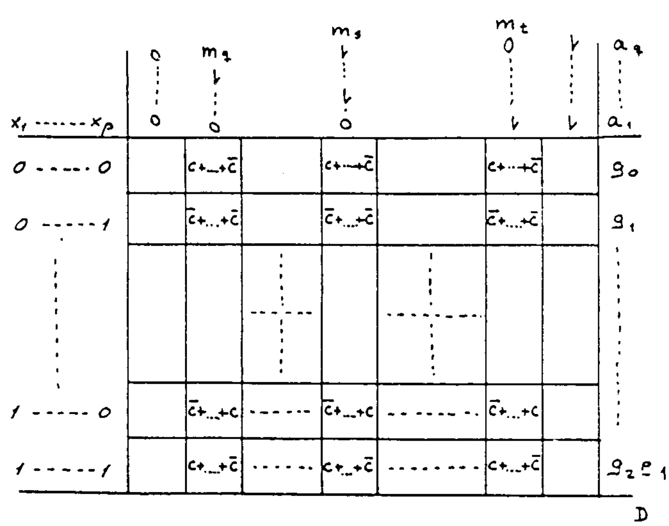

Fig. 11: El producto lógico de las columnas correspondientes a términos redundantes es cero.

Ello es siempre posible, ya que un elemento, d_i,s , de la columna s tiene una forma tal - como:

$$m_i(x_j) \to \overset{F_1}{\downarrow}\!C + \overset{F_2}{\downarrow}\!C + \ldots + \overset{F_n}{\downarrow}\!C$$

y siempre es posible definir una asignación de valores a adoptar por F_1, F_2,... F_n para el minterm $m_s(a_k)$ tal que haga "0" esa suma lógica.

El resultado es: "Los términos redundantes - no introducen incompatibilidades". Así pues, la CNS para la existencia de solución, en el caso de funciones incompletas, es que "en cada columna correspondiente a término no re-- dundante haya, al menos, un elemento "0"".

Queremos saber ahora cuáles y cuántas son -- las asignaciones permitidas, es decir, las - que podemos adoptar para no introducir incompatibilidad en el sistema.

Para una columna dada de D correspondiente a un término redundante (Fig. 12), por cada -- elemento se puede adoptar una asignación, y sólo una, que lo haga nulo. Como hay 2^p elementos podría haber como máximo 2^p posibles asignaciones diferentes (si $p \leq n$). Obviamente el número máximo de asignaciones posible será 2^n. Pero, en general, será inferior a 2^n porque no todas serán permitidas.

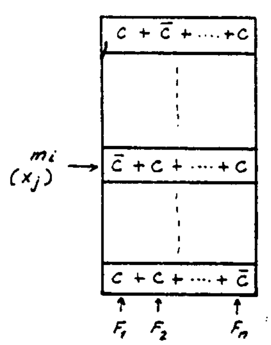

Fig. 12: Existe una única asignación que anula a cada columna correspondiente a un término redundante.

Ahora hay que tener en cuenta que todas y cada una de las asignaciones que hacen "0" al producto lógico de los elementos de una co-- lumna correspondiente a un determinado término redundante hacen también "0" a los demás. Esto se deduce de la igualdad que tienen los elementos de la misma fila en todas las co-- lumnas correspondientes a términos redundantes.

Supongamos que el número de asignaciones permitidas (cualquiera de ellas hace "0" al menos un elemento de la columna) es w y que - el número de términos redundantes es r. Teniendo en cuenta que dos columnas correspondientes a términos redundantes pueden hacerse "0" adoptando por separado cualquiera de las asignaciones permitidas, el número de - sistemas de ecuaciones a resolver es de w^r, sin que ninguno de ellos introduzca incompatibilidades debidas a términos redundantes.

3.3. RESOLUCION CON LA PRIMERA ETAPA FIJADA.

Se deberá cumplir la ecuación booleana (ver Fig. 13)

$$f_k\ (g_e) = F_k(x_j) \qquad \begin{array}{l} j=1,\ldots p \\ e=1,\ldots m \\ k=1,\ldots n \end{array} \qquad (7)$$

verificándose además las ecs. de ligadura:

$$g_e = G(x_j) \qquad (8)$$

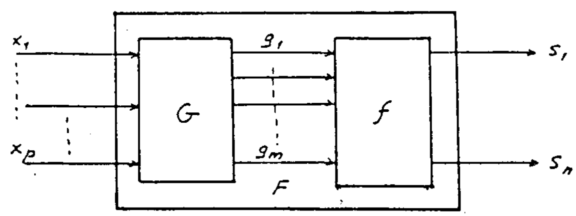

Fig. 13: Descomposición de F con la 1^a. etapa fija.

La resolución del sistema de ecuaciones (7) con las ligaduras (8) ha de proporcionar las soluciones del sistema, si existen, que se-- rán de la forma: $S_k=f_k(g_e)$.

El método desarrollado paa sintetizar la función F utilizando el tipo de descomposición indicado se basa en lo siguiente:

a) Desarrollar y expresar cada función f_k en forma canónica en función de las varia--- bles g_e.

b) Sustitución de las ecuaciones de ligadura (8) en el desarrollo mencionado.

c) Estudio de la CNS de existencia de solu-- ción.

d) Obtención de las soluciones.

Formalmente se puede escribir:

$$S_k = f_k(g_e) = \sum_{i=0}^{2^m-1} c_i^k \, m_i(g_e) \qquad (9)$$

$$S_k = F_k(x_j) = \sum_{1=0}^{2^p-1} c_1'^k . m_1(x_j) \qquad (10)$$

La CNS para la existencia de solución es:

"a) todo minterm de las variables x_j que forma parte de (10) ha de aparecer en (9) - como parte de algún minterm de las variables g_e, y

b) este minterm de las variables g_e no puede contener otro u otros minterms de las variables x_j que no aparezca en (10)".

La obtención de las soluciones cuando se verifica la CNS de existencia de solución es - fácilmente automatizable /8/ en el caso de - funciones completas. Para funciones incompletamente específicadas, basta con no considerar la construcción de las salidas para las configuraciones de entrada correspondientes a términos redundantes.

4. CONCLUSIONES.

Se ha atacado el problema de síntesis de funciones de conmutación de salida múltiple, a partir de módulos fijos cualesquiera de salida múltiple.

Se ha reducido el problema a la síntesis de una función en dos etapas separadas, siendo conocida una de ellas.

Se ha resuelto el problema conociendo la segunda etapa, tanto para funciones completas como incompletas. En el caso de no existencia de solución, se ha resuelto el problema redefiniendo el circuito de tal manera que - se aproveche al máximo el módulo fijado, es decir, que éste fabrique el máximo número posible de salidas.

Se ha resuelto el problema cuando se conoce la primera etapa, tanto para funciones completas como incompletas.

Quedan por tratar ciertos problemas importantes:

a) Cómo minimizar las funciones solución (1^a etapa o 2^a etapa) en relación al proceso iterativo de síntesis en múltiples etapas.

b) En relación con a) el estudio de los términos redundantes ligados.

c) Cómo modificar el circuito en el caso de incompatibilidad cuando se conoce la primera etapa, para aprovechar al máximo el módulo fijo. Es decir, cómo construir de forma óptima en coste los minterms de -- las variables de entrada que no pueden - ser reconstruidos con la función solución a partir de las variables intermedias.

d) Obtener métodos globales que abarquen todas las posibles soluciones para imponer el criterio de minimización de coste más conveniente, en lugar de estudiar una a una todas las soluciones y seleccionar -- "a posteriori" la mejor.

5. BIBLIOGRAFIA.

/1/ ASHENHURST, R. "Non-disjoint Decomposition", Harvard Computation Lab. Report n°BL-4, Sec. IV (1953).

/2/ BROWN, F.M. "Reduced Solution of Boolean Equations", IEEE Trans. Comp., C-19, --- (Oct. 1970). pp. 976-9P1.

/3/ CERNY, E. & M.A. MARIN, "A Computer Algorithm for the Synthesis of Memoryless Logic Circuit", IEEE Trans. Compt.C-5 (mayo 1974).

/4/ CURTIS, H.A. "Simplefied Decomposition - of Boolean Equations" IEEE Trans.Comput. C-25.

/5/ DESCHAMPS, J.P., "Digital Processes", 1, 123-140 (1975)

/6/ DORMIDO, S., CANTO, MA. A. y VAQUERO, A. "Aplicación de la descomposición funcional a la síntesis de funciones múltiples con módulos lógicos universales tipo multiplexor", Real Soc. Esp. Fís. y Quim. XVII Reunión Bienal, libro G, Simp. 12, comunicación 37, Sept. 1980.

/7/ VAQUERO, A., FUSTER, A., MARIN, M.A. y -- VALDERRAMA, E., "Descomposición de funciones booleanas múltiples en dos etapas, estando fijada la segunda". Revista de la - Real Academia de Ciencias Exactas, Físicas y Naturales, tomo LXXIV, cuad. IV, -- pp. 783-786 (1980).

/8/ VAQUERO, A.,MARIN, M.A. "Descomposición - de funciones booleanas múltiples en dos - etapas, estando fijada la primera" ------ XVIII Reunión Bienal de la Real Soc. Esp. de Fís. y Quím. Burgos 29 Sept.-3 Oct. -- 1980. Comunicación 12-39.

/9/ MELLADO, M., "Síntesis de funciones lógicas con operadores de intersección (Unión y "O" exclusiva". III Congreso Nacional - de Informática y Automática, Madrid, 1975.

/10/ MUKHOPDHYAY, AMAR & GREG SMITH, "Minimization of Exclusive or and Logical Equivalence Switching Circuits", IEEE Trans. -- Comput., vol. C-19, n°2, 1970.

/11/ SVOBODA, A. "A Algorithm for Solving Boolean Equations" IEEE Trans. Comput. Ec-12 pp. 557-558 Oct. 1963.

/12/ TAPIA, M.A. & TUCKER, J.H. "Complete Solution of Boolean Equations" IEEE Trans. -- Comput, C-29 (Julio 1980).

/13/ VALDERRAMA, E., VAQUERO, A. y AGUILO, J. "Descomposición de funciones booleanas - múltiples en dos etapas, fijando la segunda". Real Soc. Esp. de Fís. y Quím. - Actas Reunión 75 Aniversario. Madrid, 27 oct. 1978, pp. 89-90.

/14/ VAQUERO, A. "Síntesis de circuitos lógicos binarios modulares" Revista de la Real Academia de Ciencias LXXII (1978), pp.625-628.

/15/ WHITEHEAD, A.N. "Memoir on the Algebra of Symbolic Logic", Amer. J. Math., vol. 23 (1901), pp. 140-165 y 297-316.

/16/ MARIN, M.A. "Síntesis de circuitos combinacionales usando circuitos integrados" - Rev. de Automática, n°3, 1969.

/17/ IGLESIAS, M. "Descomposición de funciones booleanas múltiples incompletas, en dos - etapas, estando fijada la segunda". Memoria de licenciatura, Universidad Complutense, Facultad de Ciencias Físicas, junio 1981.