

SINTESIS DE SUMADORES BIPOLARES

E. MELLADO, E. A. RENTERIA, J. M. TARELA

La clase de códigos denominada bipolar binaria incluye un amplio conjunto de códigos binarios con ciertas propiedades comunes. Este hecho, así como la circunstancia de que ciertos conjuntos de operadores funcionalmente completos que incluyan al EXOR facilitan la síntesis de funciones, se aplica a la obtención de sumadores generalizados tanto en el sentido de que aceptan más de dos entradas simultáneas como en el sentido de que éstas pudieran pertenecer a códigos diferentes.

1. INTRODUCCION.

El estudio de las propiedades lógicas del operador unión exclusiva, suma natural o exor, constituye el propósito de este trabajo, tanto desde el punto de vista de la síntesis óptima de circuitos como de la aplicación de los resultados obtenidos a la realización de operaciones aritméticas rápidas.

2. REPRESENTACION DIGITAL DE NUMEROS.

Para realizar el estudio de circuitos aritméticos es necesario considerar los distintos métodos de representación de números, en especial aquellos que abarquen tanto a los positivos como a los negativos.

De acuerdo con esto, los sistemas de numeración más extendidos orientados a la realización de operaciones aritméticas son los siguientes: signo y cantidad, complemento a dos, complemento a uno, negabinario, dígito signado, y ya en el campo de los sistemas multinivel, el ternario simétrico, /1/, /2/, /3/.

Del estudio de estos códigos es posible establecer uno que denominamos binario bipolar, /4/, en el cual un número A es representado

por la palabra $[(S_n, a_n), (S_{n-1}, a_{n-1}), \dots, (S_0, a_0)]$ con la asignación:

$$A = \sum_{i=0}^n (-1)^{S_i} a_i 2^i \quad (1)$$

siendo (S_i, a_i) variables booleanas, e interpretando (1) con el usual sentido algebraico. A veces puede utilizarse para (1) la forma más compacta

$$(-1)^{S_i} a_i = b_i \quad (2)$$

donde b_i es la variable bipolar $b_i \in \{-1, 0, +1\}$. Pues bien, puede decirse que los métodos de representación numéricos mencionados anteriormente (signo y cantidad, complemento a 2, negabinario, etc.) son casos particulares o subconjuntos distintos del que se deduce de (1) y que hemos denominado binario bipolar; en efecto sea la palabra $[x_n, x_{n-1}, \dots, x_1, x_0]$ y consideremosla en dichos códigos; será:

Signo y cantidad:

$$\begin{aligned} S_n &= a_n = 0 \\ S_{n-1} &= S_{n-2} = \dots = S_0 = x_n \\ a_i &= x_i \quad \forall i < n \end{aligned}$$

Complemento a dos:

- M. Mellado - Dpto. Fisica Industrial. -Univ. Complutense de Madrid - Madrid
- E. A. Renteria - Dpto. Automática e Informática - Univ. del Pais Vasco - Ciudad Universitaria -Lejona (Vizcaya)
- J. M. Tarella - Dpto. Electricidad y Electrónica - Univ. del Pais Vasco - Ciudad Universitaria -Lejona (Vizcaya)
- Article rebut el Setembre de 1981.

$$S_n = 1$$

$$S_{n-1} = S_{n-2} = \dots = S_0 = 0$$

$$a_i = x_i \quad \forall i \leq n$$

Negabinario:

$$S_0 = S_2 = S_4 = \dots = 0$$

$$S_1 = S_3 = S_5 = \dots = 1, \text{ y } a_i = x_i \quad \forall i \leq n$$

Dígito signado:

$$S_i = \sum_{j=0}^i x_j, \text{ y } a_i = x_i \quad \forall i \leq n$$

A nuestro juicio el código bipolar sugiere - importantes caminos de investigación de los cuales los más inmediatos son los que se detallan a continuación:

- El desglose establecido al separar el signo podría permitir realizaciones circuitales ventajosas. Por ello es preciso estudiar las características de una aritmética bipolar.

- Los códigos concebidos para transmisión de información son superabundantes con el fin de utilizar este hecho para la detección y - posible corrección de errores. Pues bien, es fácil intuir la posibilidad de utilizar las configuraciones redundantes de (1) en este - sentido.

- Otra línea a seguir, partiendo precisamente de la superabundancia del código bipolar, es la investigación en el campo de la tolerancia de fallos, uno de cuyos propósitos es el de hacer utilizables aquellos circuitos - integrados parcialmente defectuosos.

- Por último, dado que en este código para - cada posición pueden modificarse tanto el -- bit de peso como el signo, y éstos pueden -- ser utilizados en la adaptación rápida de señales, así las aproximaciones a un valor dado podrían ser más rápidas.

3. DESARROLLO, TRANSFORMACIONES Y SINTESIS - OPTIMA DE FUNCIONES BOOLEANAS EXPRESADAS EN TERMINOS DEL OPERADOR EXOR.

Es fácil demostrar que el conjunto de operadores {EXOR, INTERSECCION} (en lo sucesivo { \oplus, \cdot }) es funcionalmente completo y verifica las siguientes propiedades:

P1. - Cada operador posee un único elemento neutro (respectivamente (0,1)).

P2. - Cada operador verifica la propiedad -- conmutativa.

P3. - La intersección es distributiva sobre el exor.

P4. - Para cada A existe un único \bar{A} tal que $A \oplus \bar{A} = 1$ y $A \cdot \bar{A} = 0$.

P5. - Anulación, $A \oplus A = 0$.

Dadas n variables, cada término intersección formado con las mismas recibe el nombre de - pseudominterm, y cada término unión recibe el nombre de pseudomaxterm.

Pues bien, las propiedades anteriores junto con los teoremas que de ellas pueden deducirse, y que se muestran en /4/, sirven de apoyo en la obtención de las cuatro formas canónicas siguientes: exor de minterms, exor de maxterms, exor de pseudominterms y exor de pseudomaxterms. Todas ellas se resumen a continuación:

a) Exor de minterms:

$$f(x_1, \dots, x_n) = \sum_{i=0}^{2^n-1} m_i f_i \quad (3)$$

donde m_i es cada uno de los posibles términos producto que se pueden construir con las n variables tomadas complementadas o no, y f_i son los coeficientes constantes del desarrollo de f. A esta forma se llega dado que la unión y el exor pueden intercambiarse cuando afectan a conjuntos disjuntos.

b) Exor de maxterms:

$$f(x_1, \dots, x_n) = \sum_{i=0}^{2^n-1} (M_i + F_i) \quad (4)$$

donde M_i es cada uno de los posibles términos suma que se pueden construir con las n variables tomadas complementadas o no, y F_i son los coeficientes constantes que definen a f de este modo. A este se llega -- complementando término a término en la expresión anterior, ya que al complementar un número par de términos relacionados por el exor no se modifica la función.

c) Exor de pseudominterms:

$$f(x_1, \dots, x_n) = \sum_{i=0}^{2^n-1} \alpha_i a_i \quad (5)$$

donde α_i es cada uno de los términos producto que se pueden construir con cualquier subconjunto de las n variables tomadas sin complementar, con $\alpha_0=1$; y a_i las constantes que definen este desarrollo. Se puede llegar a éste partiendo del desarrollo en minterms; un minterm con p variables complementadas da lugar a 2^p pseudominterms.

d) Exor de pseudomaxterms:

$$f(x_1, \dots, x_n) = \sum_{i=0}^{2^n-1} (\beta_i + A_i) \quad (6)$$

donde β_i es cada uno de los términos suma que se pueden construir con cualquier subconjunto de las n variables tomadas sin complementar, con $\beta_0=0$; y A_i las constantes que definen este desarrollo. Se llega a este desarrollo partiendo del anterior supuesto que se hubiera realizado sobre las variables complementadas y como se hizo en el caso de los maxterms, complementando todos los términos.

Los pseudominterms α_i y los pseudomaxterms β_i se pueden identificar fácilmente por comparación con los minterms m_i y los maxterms M_i sin más que omitir las variables complementadas.

En otro orden de cosas es de notar que todo lo mencionado hasta aquí para funciones simples es extensible a funciones vectoriales.

Obtenidos los distintos desarrollos se han encontrado las siguientes transformaciones:

De minterms a maxterms:

$$|F_i| = |I|. \left| \bar{f}_{2^n-1-i} \right| \quad (7)$$

donde $|I|$ es la matriz identidad.

De minterms a pseudominterms:

$$|a_i| = |T_n|. |f_i| \quad (8)$$

siendo T_n una matriz que se puede obtener mediante la relación recursiva:

$$|T_j| = \begin{vmatrix} T_{j-1} & 0 \\ T_{j-1} & T_{j-1} \end{vmatrix} \quad \text{y} \quad |T_1| = \begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix} \quad (9)$$

De minterms a pseudomaxterms:

$$|\bar{A}_n| = |T_n|. |f_{2^n-1-i}| \quad (10)$$

Puesto que las matrices $|I|$ y $|T_n|$ son auto-inversas es posible obtener cualquier otra relación.

En el caso de desear desarrollos sobre conjuntos de variables en las que aparezca cualquier número de ellas complementado, las matrices de transformación son las mismas, únicamente se debe recordar la forma en la que se han asignado los subíndices.

Una vez obtenidas estas formas canónicas es preciso establecer un criterio a la hora de realizar minimizaciones de las funciones construidas sobre $\{0,1\}$.

Si se supone que cada una de estas operaciones se realiza en un nivel y se utiliza el criterio de óptimo según el cual se desea una implementación con el mínimo número de puertas, y entre éstas aquellas que poseen menos entradas es de fundamental importancia utilizar aquellos teoremas que permitan lograr este objetivo.

Cuando la función esté expresada en exor de minterms tales teoremas son, de una parte

$$T1: \quad x_1 x_2 \oplus x_1 \bar{x}_2 = x_1 \quad (11)$$

y de otra el postulado de anulación, P5. La aplicación de estas reglas puede hacerse sobre mapas de Karnaugh o, en forma tabular a través de un algoritmo exhaustivo.

Si la función está en exor de pseudominterms las correspondientes relaciones son P5 y

$$T2: \quad x_1 x_2 \oplus x_1 = x_1 \bar{x}_2 \quad (12)$$

en este caso la función puede ser minimizada sobre mapas de Karnaugh modificados o también en forma tabular como en el caso anterior.

4. DESARROLLO DE UN SUMADOR GENERALIZADO. APLICACIONES.

Utilizando un algoritmo que se describe a -- continuación se han diseñado los módulos de suma que permiten la realización de ésta de diferentes formas. En todos los casos, como se verá, la estructura es de tipo iterativo.

Siempre que se desee realizar una suma es posible calcular en una primera fase la suma -- por un lado y el arrastre por otro para cada columna; si los sumandos tienen n bits se -- tendrán n resultados suma y n resultados --- arrastre, y de los n resultados suma el de -- menor peso es definitivo. En la siguiente fa se o etapa se repite el proceso con los n-1 resultados suma de mayor peso y los n-1 ---- arrastres de menor peso como sumandos, reser vándose el resultado arrastre de mayor peso para la etapa n+1. Al cabo de n etapas se -- tienen n salidas suma de caracter definitivo, el resultado de mayor peso se obtiene de sum ar los arrastres abandonados en cada etapa.

Del carácter iterativo del algoritmo así como del hecho de que, dentro de cada etapa, -- para cada columna se realice la misma fun--- ción se desprende su estructura modular, lo cual revierte en la economía del dispositivo resultante.

Se muestran en este resumen las funciones -- que se obtienen para el caso de que un suman do pertenezca a un código bipolar y el otro sea binario natural. Así los módulos que rea lizan estas funciones y la disposición espacial en el caso que el código de salida sea igual al del sumando bipolar se muestra en -- la figura 1. Cuando el código de salida es -- en complemento a dos el resultado es el de -- la figura 2.

Las versiones secuenciales se obtienen direc tamente a partir de las tablas de verdad de los módulos diseñados, fijando en éstas las variables de entrada y de salida que se de-- sea hacer corresponder con los estados inter nos del sistema.

En el caso del sumador elemental con signo, cuya tabla es la de la figura 3a, realizando la asignación:

Estado actual:

A_0 para el sumando $b_i=0$

A_1 para el sumando $b_i=1$

Estado siguiente:

A_0 si $c_i=0$

A_1 si $c_i=1$

se llega a la tabla de estados de la figura 3b. Realizada la síntesis sobre F.F. J.K. -- el módulo que realiza la operación deseada es el de la figura 4 y el esquema para tres bits es el de la figura 5.

Una aplicación de los módulos anteriores se presenta en el diseño de sumadores múltiples, de interés en la realización de multiplicado res.

Una forma de interpretar el algoritmo en este caso consiste en tomar los sumandos de -- dos en dos, llevar éstos sobre los corres-- pondientes sumadores. Una vez atravesado el primer nivel de puertas se tienen

$$p = \text{ent}\left(\frac{m+1}{2}\right) \quad (15)$$

resultados suma para cada columna y otros -- tantos resultados arrastre, la columna de -- los p resultados suma de menor peso irá so-- bre el segundo bloque de sumadores.

En el segundo nivel de puertas se repite el proceso para $2p$ sumandos, número de sumandos que se mantiene en todos los niveles de operación de este bloque de sumadores. Para el segundo bloque se tendrá una suma de p suman dos con n+2 bits, en el caso más general, su puesto que inicialmente los sumandos tuvie ran n bits.

El esquema general es el de la figura 6.

Los módulos diseñados son utilizables para -- realizar sumas en otros códigos.

Dado que como se ha dicho la salida se puede obtener en complemento a dos en código bipolar (o en un subconjunto del mismo) es inmediato el uso del sumador como decodificador en estos casos.

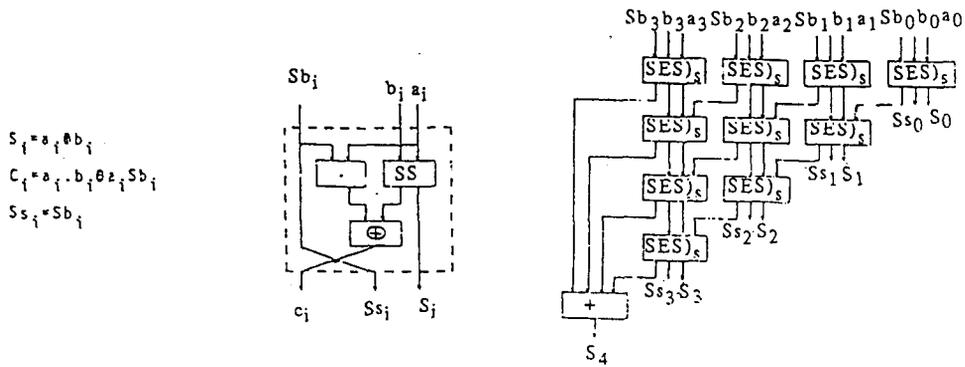


Fig. 1: a) ecuaciones. b) celda elemental. c) sumador total

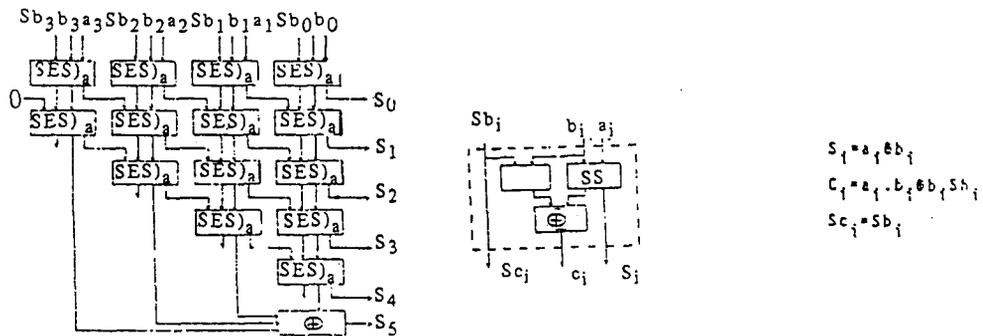


Fig. 2: a) ecuaciones. b) celda elemental. c) sumador total

a_i	b_i	S_{i-1}	S_i	c_i	S_{i+1}
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	1	0	0	0

a)

C_i	C_{i+1}			
	$\bar{a}_i \bar{b}_i$	$\bar{a}_i b_i$	$a_i \bar{b}_i$	$a_i b_i$
A_0	A_0/v	$A_0/0$	$A_0/1$	$A_0/1$
A_1	A_0/c	$A_1/1$	A_1/l	$A_0/0$

Fig. 3: a) tabla de la celda de la Fig. 2b. b) tabla de estados

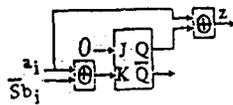


Fig. 4: celda elemental SES)a

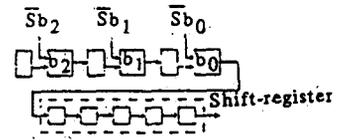


Fig. 5: versión secuencial del sumador total.

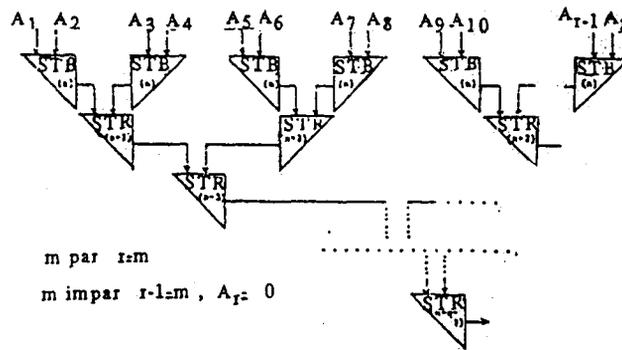


Fig. 6: sumador múltiple.

5. CONSECUENCIAS.

En trabajos previos se demuestra que funciones con determinado grado de simetría, el caso de la suma en base 2 es un ejemplo, la utilización de conjuntos de operadores funcionalmente completos que contengan al EXOR puede dar lugar a resultados "más simples" - en algún sentido.

La representación bipolar aquí estudiada y la realización de circuitos en la misma permite resultados operativos tan notables como los dos siguientes:

- diseñar circuitos que realicen operaciones sobre entradas en distintos códigos.
- diseñar circuitos generalizados que den su salida en un código dado independiente del de entrada (o de los de entrada).

Pues bien, uno de los propósitos del trabajo ha sido introducir esta representación generalizada.

Por otra parte dado que en el diseño de este tipo de circuitos aparecen funciones del tipo mencionado anteriormente, la síntesis de los mismos es de esperar que sea más eficaz utilizando conjuntos de operadores que contengan al EXOR. Esta ha sido la razón por la cual para implementar todos los circuitos de este trabajo se han utilizado conjuntos de operadores que contienen el EXOR como uno de ellos. En la referencia 4 se describen procedimientos de minimización dentro de estos conjuntos de operadores.

Finalmente deseamos resaltar que los resultados obtenidos hasta el momento nos sugieren las líneas de investigación siguientes:

- Fijados los signos de cada posición, es de cir para subconjuntos del código binario bipolar de interés, obtener las versiones secuenciales simplificadas.

- En este mismo sentido estudiar el costo de los dispositivos de suma en todos los casos posibles.

- Partiendo del estudio de los sumadores múltiples, profundizar en el estudio de multiplicadores y analizar la reducción del costo para códigos fijos.

- Completar el estudio de operaciones básicas realizando el estudio del divisor.

6. BIBLIOGRAFIA.

/1/ OBERMAN. "Digital for binary arithmetic", Mc. Millan, 1969.

/2/ D.E. KNUTH. "The art of computer programming", Vol. 21 Seminumerical algorithms, Addison-Wesley, 1969.

/3/ DAO T.T.-DARIOM-GROSSARTC. "Complex number arithmetic with odd valued logic". IEEE Trans. Comp., Vol. C-29 no 7, July 1.980.

/4/ E. A. RENTERIA. "Circuitos aritméticos sobre códigos unipolares y bipolares", Memoria de Grado de Doctor en Ciencias Físicas. Universidad del País Vasco. Facultad de Ciencias, Bilbao, 1.981.

/5/ A.D. FRIEDMAN-P.R. MENON. "Theory and design of switching circuits". Pitman, 1.977.