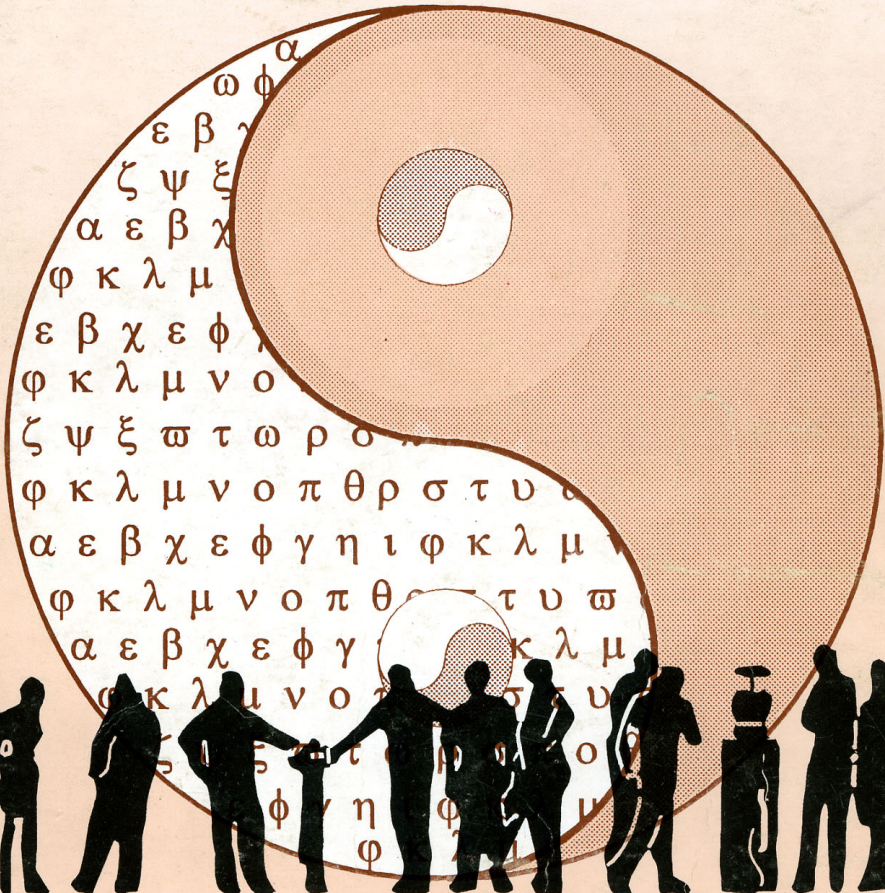


Ingeniería

Revista de la Universidad de Costa Rica
JULIO/DICIEMBRE 1992 VOLUMEN 2 Nº 2

620
IN



INGENIERIA

Revista Semestral de la Universidad de Costa Rica

Volumen 2 Julio-Diciembre 1992 Número 2

DIRECTOR

Rodolfo Herrera J.

EDITOR

Victor Herrera C.

CONSEJO EDITORIAL

Víctor Hugo Chacón P.

Ismael Mazón G.

Domingo Riggioni C.

CORRESPONDENCIA Y SUSCRIPCIONES

Editorial de la Universidad de Costa Rica

Apartado Postal 75

2060 Ciudad Universitaria Rodrigo Facio

San José, Costa Rica

CANJES

Universidad de Costa Rica

Sistemas de Bibliotecas, Documentación e información

Unidad de Selección y Adquisiciones-CANJE

Ciudad Universitaria Rodrigo Facio

San José, Costa Rica

Suscripción anual:

Costa Rica: ₡750,00

Otros países: US \$20.00

Número suelto:

Costa Rica: ₡500,00

Otros países: US \$10.00



DETECCION DE FALLAS CON EL ALGORITMO D.

Geovanny Delgado C.

RESUMEN

En este artículo se presenta un método sistemático para la detección de fallas en sistemas digitales combinatoriales: el algoritmo D. En la primera parte se estudian algunas definiciones sobre este algoritmo, esta sección se ha denominado Calculo D, a partir del cual se define la Cobertura Singular, los Cubos Primitivos de Falla y los cubos de Propagación de Falla. Luego se presenta el algoritmo D constituido por el Conducto-D y la Operación de Consistencia. Finalmente el trabajo se enriquece con un ejemplo y una simulación por computador.

SUMMARY

This article presents a method for detection of failures in combinational digital systems: the D-Algorithm. In first place we examine some definitions on this algorithm. This section was denominated D Calculus, then we define Singular Cover, the primitive D-cube of Failure and the Propagation D-cubes. After that we present the D-algorithm constituted for the D-drive and the Consistency Operation. Finally, this work is enhanced with an example and a computer simulation.

1. CONCEPTOS GENERALES.

El algoritmo D es uno de los métodos más importantes para la detección de fallas en sistemas digitales. Sus principales características son: permite detectar múltiples fallas, permite detectar fallas que no son del tipo $s-a-1$ ó $s-a-0$ y su principal ventaja es que no requiere representar y almacenar el circuito completo a ser analizado por tablas, fórmulas o ecuaciones. Esta última característica hace posible la aplicación del algoritmo a circuitos de múltiples niveles sin los típicos problemas de almacenamiento.

La filosofía del algoritmo D se basa en el hecho de que al presentarse una falla se genera una señal interna errónea en el circuito, la cual se propagará hasta un terminal de salida, en caso que el sistema digital sea observable. Además deberá existir una o más palabras del alfabeto de entrada que estimulen esta señal errónea, es decir, que impongan las condiciones para la existencia de la señal errónea en el sistema digital.

De esta manera, para una falla en un punto interno en el circuito, deberá determinarse si esta puede propagarse a los largo de alguna o algunas rutas hasta una salida (circuito observable) y si existe una palabra de entrada que genere esta señal (circuito irredundante).

Al final del proceso, obviamente, el resultado de la aplicación del algoritmo D será una prueba o conjunto de pruebas que permitirá detectar una falla en particular en un sistema digital.

2. ANALISIS EN VARIABLE D.

2.1 ASPECTOS INTRODUCTORIOS.

El cálculo D explica y justifica el procedimiento de generación/propagación de una falla particular en un sistema digital. Las siguientes observaciones preliminares deben contemplarse:

i) Una variable D en un punto de un circuito digital representa una señal que normalmente se encuentra en 1 y que, debido a la falla, está en cero.

ii) Una variable D' representa una señal que normalmente está en 0 pero que, debido a la falla, se encuentra en 1.

2.2 CALCULO D.

Para estudiar sistemáticamente la propagación de fallas se establece primero que una línea del circuito digital puede tomar cualquier valor lógico: 0, 1, d, D , D' , donde d representa un estado no importa.

Con el objetivo de propagar o conducir una D o D' desde una línea con falla a las entradas y salidas del circuito, es necesario describir cada bloque (o compuerta) del circuito bajo estudio por una **cobertura singular**.

2.2.1 COBERTURAS SINGULARES

Sea la caracterización funcional de un bloque (o compuerta) dada por $f(X_1, X_2, \dots, X_n)$ la cual está

representada como una función de $(n+1)$ variables $g(X_1, X_2, \dots, X_n, X_{n+1})$, tal que, $g(a_1, a_2, \dots, a_n, a_{n+1})=1$, si y solo si, $a_{n+1}=f(a_1, a_2, \dots, a_n)$, entonces se tiene la siguiente definición:

DEFINICION 2.2.1:

El conjunto de todos los implicantes primos de la función g es llamado cobertura singular de la función f .

De tal manera que un conjunto de implicantes primos que cubren la función g representa la tabla de verdad de la función f de una manera consisa, los cuales representan su cobertura singular.

En la Tabla # 2.2.1 se muestran las coberturas singulares para las seis compuertas básicas, de dos entradas X_1 y X_2 .

TABLA #2.2.1 Coberturas Singulares de las compuertas básicas			
COMPUERTA	COBERTURA SINGULAR (SC)		
	X1	X2	f
Y	1	1	1
	0	d	0
	d	0	0
O	1	d	1
	d	1	1
	0	0	0
NOY	0	d	1
	d	0	1
	1	1	0
NOO	0	0	1
	1	d	0
	d	1	0
O-EX	1	0	1
	0	1	1
	0	0	0
	1	1	0
NO	0	-	1
	1	-	0

El proceso de generación y propagación de la falla en un circuito combinacional está determinado por dos tipos distintos de entradas llamadas: cubos primitivos de falla y cubos de propagación de falla.

2.2.2 CUBO PRIMITIVO-D DE FALLA (pdfc)

DEFINICION 2.2.2:

Un cubo primitivo-D de falla (denotado por pdfc) para una falla de un bloque es una prueba, es decir, un conjunto completamente especificado de entradas al bloque, las cuales causan que las salidas del bloque sean diferentes de su valor normal, si la falla está presente.

Por ejemplo, para la compuerta NOO, si su entrada X1 está s-a-1, X1=0, X2=0, causará que la salida f=0, si la falla está presente y f=1 si no existe falla. Esto es representado por el siguiente cubo primitivo-D:

X1	X2	f
0	0	D

El cubo primitivo-D 00D indica que para las entradas X1=0 y X2=0 la salida del circuito será 1 si no existe falla y 0 en caso de que se presente una falla.

Los cubos primitivos-D para las seis compuertas lógicas básicas se muestran en la Tabla # 2.2.2

TABLA # 2.2.2 Cubos primitivos-D de las compuertas básicas.			
COMPUERTA	CUBO PRIMITIVO-D (pdfc)		
	X1	X2	f
Y	1	1	D
	0	1	D'
	1	0	D'
O	1	0	D
	0	1	D
	0	0	D'
NOY	1	0	D
	0	1	D
	1	1	D'
NOO	0	0	D
	0	1	D'
	1	0	D'
O-EX	0	1	D
	1	0	D
	0	0	D'
	1	1	D'
NO	0	-	D
	1	-	D'

2.2.3 CUBOS DE PROPAGACION-D (pdc)

DEFINICION 2.2.3:

Un cubo de propagación-D de falla (denotado por pdc) de un bloque lógico, es una prueba, es decir, un conjunto de entradas, que causan que la salida del bloque dependa solo de una o más de sus entradas especificadas y entonces propaguen una falla desde estas entradas, hasta la salida del bloque.

Entonces si se debe propagar el efecto de una falla externa a una compuerta, deben escribirse los cubos de propagación, por ejemplo, en una

compuerta NOO, de entradas X1 y X2 y salida f, si su entrada X1 esta conectada a una falla D, se tiene el siguiente cubo de propagación-D:

X1	X2	f
D	0	D'

Aquí la interpretación de la variable D es ligeramente distinta. D puede ser 0 o 1, pero todas las D en un mismo cubo tienen el mismo valor, también D' es el complemento de D.

Los cubos de propagación para las compuertas básicas aparecen en la Tabla #2.2.3

TABLA # 2.2.3			
Cubos de propagación-D de las compuertas básicas.			
COMPUERTA	CUBO PROPAGACION-D (pdc)		
	X1	X2	f
Y	D	1	D
	1	D	D
O	0	D	D
	D	0	D
NOY	D	1	D'
	1	D	D'
NOO	D	0	D'
	0	D	D'
O-EX	D	1	D'
	D	0	D
	1	D	D'
	0	D	D
NO	D		D'

Es importante destacar la interpretación del conjunto de cubos de propagación de una compuerta, porejemplo, los cubos de propagación de la compuerta O-Exclusiva establecen que la salida y la entrada X1 tienen el mismo valor si X2=0 y tienen valor opuesto si X2=1.

2.2.4 LA INTERSECCION D.

Los cubos-D, que describen la propagación de una señal D desde una línea interna hasta la salida, son obtenidos utilizando la interseccion-D, la cual se define a continuación:

DEFINICION 2.2.4:

Sean α y β dos cubos-D, entonces la intersección-D $\alpha \cap \beta$ se define a partir de la Tabla # 2.2.4 y de las siguientes reglas:

1. $\alpha \cap \beta = \bar{\emptyset}$ (vacía) si cualquier coor-denada de la intersección es nula (\emptyset).
2. $\alpha \cap \beta = \Sigma$ (indefinida) si cualquiera de sus coordenadas es Σ .
3. $\alpha \cap \beta =$ El cubo formado por las respectivas coordenadas de la intersección si no se cumplen 1 y 2.

TABLA # 2.2.4
Reglas para la intersección-D

\cap	0	1	d	D	D'
0	0	\emptyset	0	Σ	Σ
1	\emptyset	1	1	Σ	Σ
d	0	1	d	D	D'
D	Σ	Σ	D	D	Σ
D'	Σ	Σ	D'	Σ	D'

La propagación de una señal D a lo largo de una ruta, la cual corresponde a la ruta sensibilizada puede ser obtenida analíticamente por medio de los cubos primitivo-D y de propagación-D, por medio de la intersección D. El procedimiento formal de este análisis se conoce como **Algoritmo D**.

2.3 EL ALGORITMO D.

El algoritmo D es un procedimiento sistemático para la obtención de una prueba que detecta una falla en un circuito combinacional, haciendo uso de los cubos primitivo-D y propagación-D. Este algoritmo consiste de dos partes: el Conductor-D y la Operación de Consistencia, las cuales se discuten en las siguientes secciones.

Como primer paso del algoritmo D deben obtenerse las coberturas singulares y los cubos de propagación de los bloques lógicos del circuito. Inicialmente se determinan los Cubos de Propagación de una sola entrada, es decir, aquellos cubos de los bloques lógicos en los que únicamente converge una señal D. Los Cubos de Propagación-D de múltiples entradas, es decir, aquellos en los que converge más de una señal D, son eventualmente requeridos. Esto es necesario cuando, para detectar una falla, se sensibiliza más de una ruta en el circuito y estas contienen "Fan-out" reconvergente¹. Luego de obtenerse esto se aplican los dos procedimientos que a continuación se detallan.

2.3.1 EL CONDUCTOR-D

La primera parte del algoritmo D está diseñada para conducir una señal D desde el lugar donde ocurre la falla, hasta la salida del circuito, de aquí su denominación de Conductor-D.

Para determinar el Conductor-D se inicia el proceso eligiendo un camino y un cubo primitivo-D de la falla bajo consideración. Este cubo primitivo será subsecuentemente intersecado con los cubos de propagación-D de los bloques lógicos del circuito, con el objetivo de formar una cadena conectada de coordenadas D.

¹ El "Fan-out" reconvergente se dice a a aquellos puntos del circuito de donde parten dos o más rutas a entradas de distintas compuertas.

Cualquier cubo-D que represente una prueba parcialmente formada durante el Conductor-D es llamado **cubo de prueba** y denotado por tc con un superíndice indicando el paso en que este es obtenido. Asociado a cada cubo de prueba existe un **Vector de Actividad**, el cual consiste en los números (o letras) de las líneas del circuito hasta donde una D o D' ha sido propagada en ese punto de la generación de la prueba. El vector de actividad consistirá de un único número de línea si el camino bajo sensibilización posee una única ruta.

Entonces, eligiendo un camino y un cubo primitivo-D para la falla bajo consideración, se inicia el conductor-D. El cubo primitivo-D elegido es el cubo inicial de prueba y es denotado por tc^0 . Además se denota r^0 el vector de actividad asociado con el cubo inicial de prueba. El tc^0 es entonces intersecado con uno o un conjunto de cubos de propagación-D (este número depende del número de rutas del camino, el cual es igual al número de componentes del vector de actividad), los cuales tienen una D o D' en la línea r_i^0 y una o más variables D o D' en líneas **a lo largo de la ruta** que están más cerca de la salida que la línea r_i^0 . El resultado de estas intersecciones, si al menos una no es vacía, es denotada tc^1 y el Vector de Actividad r^1 es determinado.

Tratando tc^1 como tc^0 el proceso es repetido hasta que al menos una salida del circuito esté contenida en el vector de actividad.

El Conductor-D utilizando Cubos de Propagación-D de una sola entrada puede terminar prematuramente antes de que el Vector de Actividad alcance un terminal de salida si el circuito contiene "fan-out" reconvergente. Esto ocurrirá cuando todas las intersecciones del cubo de prueba con los cubos de propagación de los bloques en el vector de actividad son vacíos (debido a alguna o algunas componentes nulas). Dos tipos de intersección vacía pueden darse. El primero de ellos es cuando se interseca un 1 y un 0 en alguna coordenada de los cubos; esto indica que la cadena D no puede ser continuada y una nueva cadena debe ser creada con algún otro

miembro del vector de actividad, si existe; en caso contrario una nueva ruta ha de ser elegida. El segundo tipo es cuando la intersección es vacía debido a que el cubo de Propagación-D tiene un 0 o un 1 en la coordenada donde el cubo de prueba tiene una D o D'. Esto indica que otra ruta en el circuito ha sido accidentalmente sensibilizada. El conductor-D puede ser continuado solo si se dispone de cubos de Propagación-D de múltiples entradas. Los cubos de propagación-D para múltiples entradas, con D o D' en las coordenadas apropiadas, pueden ser deducidos en una forma análoga a los de una sola entrada. Luego si la intersección con los cubos de propagación-D de múltiples entradas también son vacíos, debe elegirse una ruta diferente.

2.3.2 LA OPERACION DE CONSISTENCIA.

Esta operación consiste de un trazado "regresivo" desde las compuertas en la ruta hasta las entradas para especificar un número suficiente de entradas que produce la señal interna deseada.

En esta operación se toma el último cubo de prueba del conductor-D y se siguen los pasos:

1. Se interseca con cada una de las coberturas singulares de aquellos bloques lógicos cuyas salidas quedaron especificadas durante el conductor-D, generando nuevos cubos de prueba. Se sigue el orden consecutivo en la numeración del superíndice de estos. Las coberturas singulares a intersecar corresponden a bloques que no necesariamente pertenecen a la ruta sensibilizada.

2. El proceso del paso 1 se suspende si aparece una intersección vacía \emptyset en cualquier línea del circuito. Esto implica que la prueba en generación no existe.

En la operación de consistencia, con la intersección de los cubos de prueba con las coberturas singulares, los bloques lógicos son "barridos" de las salidas a las entradas. Si todas las líneas del circuito son especificadas por la operación de consistencia sin encontrar ninguna contradicción, es decir, ninguna intersección es \emptyset , entonces la **prueba** ha sido encontrada.

Debe notarse que la operación de consistencia es innecesaria si todos los valores de las líneas en el circuito quedaron especificados en el último cubo de prueba en el conductor-D.

EJEMPLO #1

Considere el circuito de la Figura #E1.1, el

cual es un circuito combinacional en el que se presenta una falla en que la línea #5 está s-a-1. Esta línea se denota, de acuerdo a la convención, con una D'. Se consideran como bloques lógicos cada una de las compuertas que constituyen el sistema. Como primer paso se determinan las coberturas singulares de los bloques lógicos, estas aparecen en la Tabla # E1.1; luego se determinan los cubos de propagación de los bloques lógicos,

FIGURA # E1.1
CIRCUITO DE PRUEBA DEL EJEMPLO 1

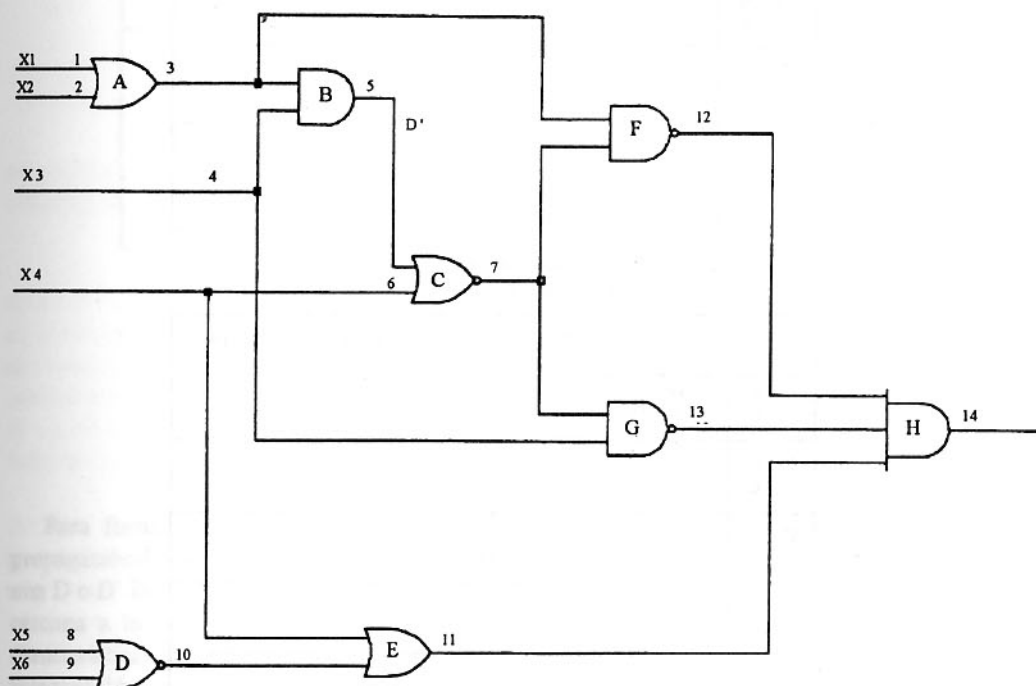


TABLA # E1.1
Coberturas Singulares del Ejemplo #1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A a b c	1 d 0	d 1 0	1 1 0											
B d e f			1 0 d	1 d 0	1 0 0									
C g h i					0 1 d	0 d 1	1 0 0							
D j k l							0 1 d	0 d 1	1 0 0					
E m n o						1 d 0			d 1 0	1 1 0				
F p q r			0 d 1				d 0 1			1 1 0				
G s t u				0 d 1			d 0 1				1 1 0			
H v x w y										1 0 d d	1 d 0 d	1 d d 0	1 d 0 0	1 0 0 0

TABLA # E1.2
Cubos de propagación del Ejemplo #1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A a _d b _d	0 D	D 0	D D											
B c _d d _d			D 1	1 D	D D									
C e _d f _d					D 0	0 D	D' D'							
D g _d h _d								D 0	0 D	D' D'				
E i _d j _d						0 D				D 0	D D			
F k _d l _d			D 1				1 D					D' D'		
G m _d n _d				D 1			1 D					D' D'		
H o _d p _d q _d											D 1 1	1 D D	1 1 D	D D D

TABLA #E1.4

CONDUCTOR D Y OPERACION DE CONSISTENCIA

	Cubo de Prueba															V.A	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14		
Cond-D	tc^0			0	1	D'											5
	$tc^1=tc^0 \wedge e_d$			0	1	D'	0	D									7
	$tc^2=tc^1 \wedge n_d$			0	1	D'	0	D						D'			13
	$tc^3=tc^2 \wedge q_d$			0	1	D'	0	D				1	1	D'	D'		14
Opera.	$tc^4=tc^3 \wedge p$			0	1	D'	0	D			1	1	D'	D'			
	$tc^5=tc^4 \wedge n$			0	1	D'	0	D		1	1	1	D'	D'			
Consis.	$tc^6=tc^5 \wedge j$			0	1	D'	0	D	0	0	1	1	1	D'	D'		
	$tc^7=tc^6 \wedge c$	0.	0	0	1	D'	0	D	0	0	1	1	1	D'	D'		

Siguiendo un proceso análogo se completa la operación de consistencia hasta tc^7 . De aquí se obtiene que una prueba para determinar la falla de la línea 5 s-a-1 es $X1=0$, $X2=0$, $X3=1$, $X4=X5=X6=0$. Note además que la salida es 0 cuando la falla no está presente y 1 en caso de falla.

Para finalizar el ejemplo se ha realizado una simulación del circuito, haciendo uso del CAD para ingeniería eléctrica denominado EE-DE-SIGNER de la compañía Visionics Corp. En la Figura #E1.2 aparece el resultado de la simulación del circuito libre de fallas, se muestran en esta figura las señales de entrada al circuito como una secuencia de palabras y la señal de salida; al llegar la secuencia de entrada a la combinación 000100 la salida genera una espiga de decodificación y luego retorna al estado cero. En la Figura # E1.3 aparece el diagrama temporal resultante de la simulación del circuito cuando la falla en la línea 5 está presente, en este caso la salida, bajo la secuencia 000100 se queda en alto, este es un resultado esperado, pues en el cubo de prueba tc^3 la línea de salida se encuentra con una D', indicando que en el circuito normal la salida es cero y en el circuito con falla es uno.

CONCLUSIONES

El Algoritmo D es un método sistemático para la detección de fallas en sistemas digitales combinatoriales.

Su principal característica es que no presenta problemas de almacenamiento en memoria, ya que no requiere almacenar el circuito en ecuaciones booleanas, proposiciones literales o tablas de falla, como en otros métodos, sino únicamente la información de la topología de sus bloques lógicos.

Basados en el algoritmo D se han desarrollado dos programas en alto nivel: el DALG-II y el TEST-DETECT. El DALG-II calcula una prueba para detectar una falla en circuitos combinatoriales grandes, mientras que el TEST-DETECT encuentra todas las fallas detectadas por una prueba dada. En los dos casos el almacenamiento en memoria es mínimo, ya que, los pdcf y pdc son calculados "en línea" en la medida en que se requieren, esto permite además que los programas puedan manejar caminos de múltiples rutas.

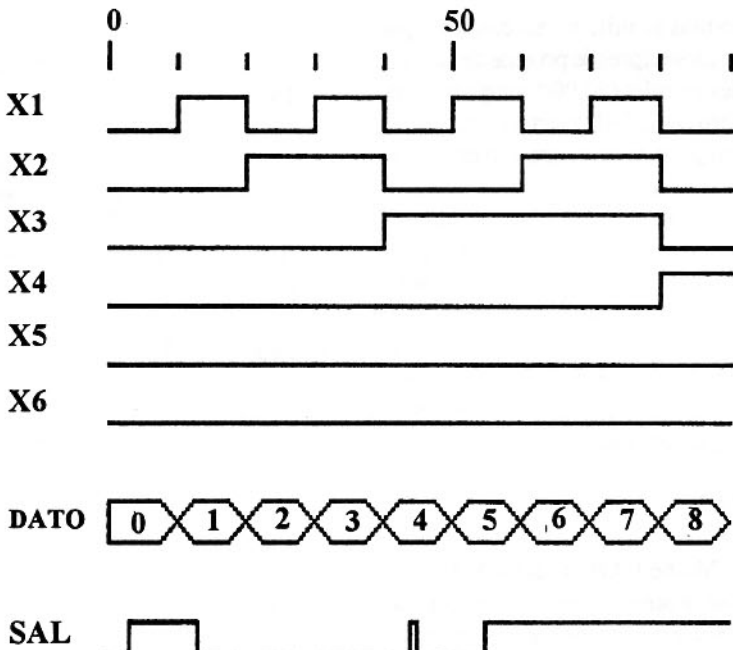


FIGURA #E1.2

DIAGRAMA TEMPORAL DEL CIRCUITO SIN FALLA

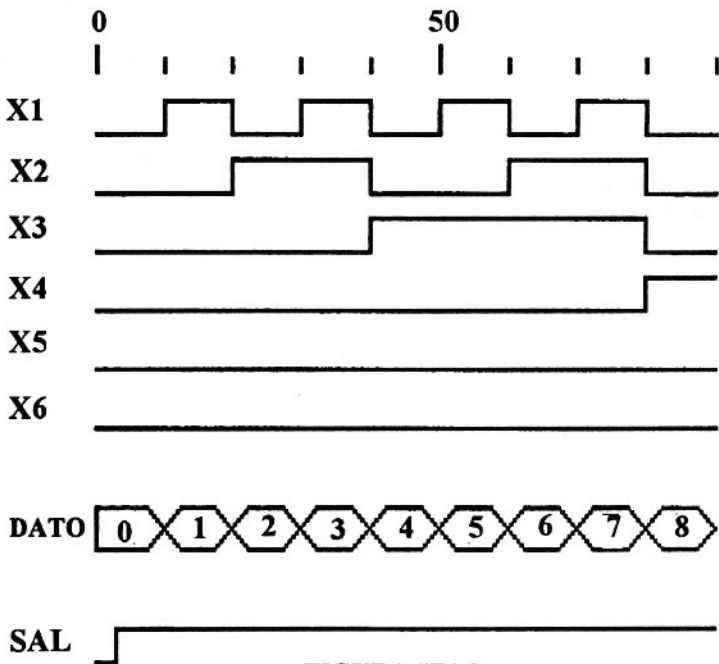


FIGURA #E1.3

DIAGRAMA TEMPORAL DEL CIRCUITO CON FALLA

Estos programas se utilizan en conjunto para la generación de los vectores de pruebas de circuitos combinatoriales grandes (50,000 compuertas, por ejemplo); primero DALG-II determina una prueba para una falla en particular, luego con este vector de prueba el TEST-DETECT indica todas las posibles fallas que se pueden detectar con este vector de prueba. A partir de esta información es posible determinar el porcentaje de cobertura de fallas en un paso dado. Para el caso de sistemas grandes generalmente sólo es posible de alcanzar hasta un 90% o un 95% de la cobertura total de fallas, en virtud de que el restante 5% llevaría demasiado tiempo cubrirlo.

REFERENCIAS

- 1) Solana, J.M., Michell, J.H., Pracho, S. *Elimination algorithm: a method for fault diagnosis in combinational circuits based on an effect-cause analysis.* (J.M. Solana, J.A. Michell, S Bracho.) En IEE Proceedings, Volumen 133, Número 1, Enero 1986. pp-31-44.
- 2) Sangwine. S.J. *Deductive fault diagnosis in digital circuits: a survey.* IEE, Proceedings, Volumen 136, Número 6, Noviembre 1989. pp-496-504.
- 3) Bhattacharya B.B. y otros *Testable design of RMC networks with universal test for detecting stuck-at and bridging faults.* IEE Proceedings, Volumen 132, Número 3, Mayo 1985. pp-154-162.
- 4) Damaria, T. y Karpovsky, M. *Detection of stuck-at and bridging faults in Reed-Muller canonical (RMC) networks.* En IEE Proceedings, Volumen 136, Número 5. Setiembre 1989. pp-430-433.
- 5) Gupta, B., Malaiya, Y.K. Min, Y., Rajsuman, R. *On designing robust testable CMOS combinational circuits.* En IEE Proceedings, Volumen 136, Número 4, Julio 1989. pp-329-338.
- 6) Bate, J.A., Miller, D.M. *Exhaustive testing of stuck-open faults in CMOS combinational circuits.* En IEE Proceedings, Volumen 135, Número 1, Enero 1988.
- 7) Lee, Samuel. *Modern Switching Theory and Digital Design.* Editorial Prentice-Hall, Iedición. New Jersey, 1978.
- 8) Friedman, Arthur, Menon, Premachandran *Fault Detection in Digital Circuits.* Edit. Prentice-Hall. New Jersey, 1971.
- 9) Lala, Parag. *Fault Tolerant and Fault Testable Hardware Design.* Edit. Prentice-Hall International; Londres, 1985.