

UNIVERSIDAD DE GRANADA
FACULTAD DE CIENCIAS

T.P. 10/107



Departamento de Electrónica y Tecnología de Computadores

**Test de circuitos digitales mediante
técnicas espectrales basadas en una
nueva transformada**



TESIS DOCTORAL

Julio Ortega Lopera

Granada, 1990



Biblioteca Universitaria de Granada



01533978

9

UNIVERSIDAD DE GRANADA

Facultad de Ciencias

Fecha 9-10-90

ENTRADA NUM. 1552

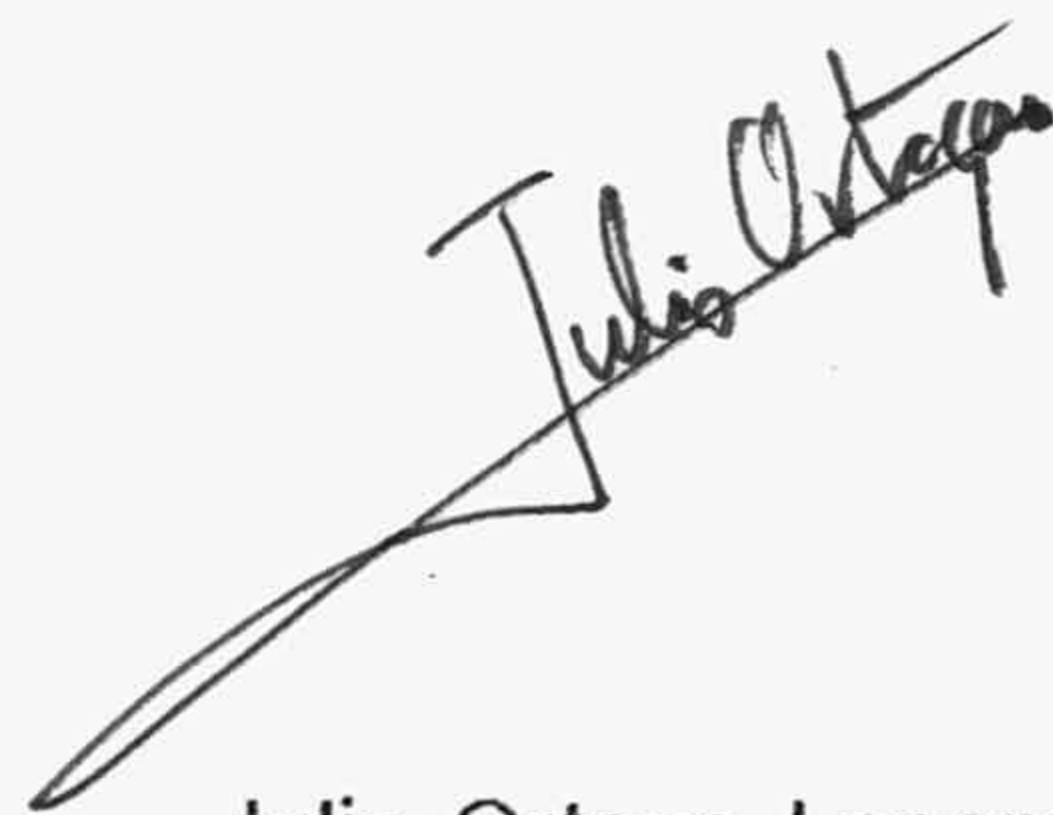
BIBLIOTECA
FACULTAD DE CIENCIAS
GRANADA

Estante _____
Tabla _____
Núm. _____

T
13
129

TEST DE CIRCUITOS DIGITALES
MEDIANTE TÉCNICAS ESPECTRALES
BASADAS EN UNA NUEVA
TRANSFORMADA

TESIS DOCTORAL



Julio Ortega Lopera

BIBLIOTECA UNIVERSITARIA
GRANADA

Nº Documento 619657675
Nº Copia 121203908



Departamento de Electrónica y Tecnología de Computadores
Universidad de Granada

Granada, 23 de Noviembre 1990

D. Antonio Lloris Ruíz,
Catedrático de Electrónica,

y

D. Alberto Prieto Espinosa,
Profesor Titular de Electrónica,
ambos del Departamento de Electrónica y
Tecnología de Computadores, de la
Universidad de Granada.

CERTIFICAN:

Que la memoria titulada: "Test de circuitos digitales mediante técnicas espectrales basadas en una nueva transformada" ha sido realizada por D. Julio Ortega Lopera bajo nuestra dirección, en el Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada. Esta memoria constituye la tesis que D. Julio Ortega presenta para optar al grado de Doctor en Ciencias Físicas.

Granada a 8 de Octubre de 1990.



Fdo. Antonio Lloris Ruíz
Codirector de la Tesis.



Fdo. Alberto Prieto Espinosa
Codirector de la Tesis.

A Carmela y a mi familia.

Mi agradecimiento a quienes, de un modo u otro, han hecho posible la realización de esta tesis, especialmente a los profesores Antonio Lloris y Alberto Prieto por sus consejos, apoyo y confianza. Al profesor S.L.Hurst por la orientación recibida, las posibilidades de consulta bibliográfica que me ofreció y su hospitalidad. Al profesor Francisco J. Pelayo con quien discutí algunos de los aspectos de este trabajo. A los compañeros del grupo de investigación de Circuitos y Sistemas para el Procesamiento de la Información y del Departamento de Electrónica y Tecnología de Computadores por la ayuda recibida y el grato ambiente de trabajo.

PROLOGO

Los sistemas electrónicos, capaces de almacenar y procesar grandes cantidades de datos e información con velocidad, precisión y fiabilidad que superan con mucho a las capacidades humanas, han introducido cambios drásticos en casi todas las actividades de la sociedad humana: propagación e intercambio de información, instrumentación científica, modelado y control de sistemas, producción agrícola e industrial, administración, educación e investigación, medicina y servicios sociales, etc. Una gran parte de la responsabilidad del uso generalizado de la electrónica y de su versatilidad corresponde al desarrollo de las tecnologías de integración LSI (Large Scale Integration) y VLSI (Very Large Scale Integration). La importancia fundamental de la integración radica en su capacidad inherente para reducir los costes de fabricación de los circuitos electrónicos, consiguiendo además incrementar su fiabilidad y sus prestaciones. La reducción de costes se debe al empaquetamiento de cada vez más circuitos en una unidad de fabricación, el chip, y a la automatización de la mayor parte del proceso de producción. La base para el incremento de prestaciones se encuentra en la reducción de las dimensiones de los circuitos, pues la disminución en los retardos que supone contribuye a incrementar su velocidad de operación. La fiabilidad de los circuitos aumenta al reducirse el número de puntos de soldadura necesarios.

Dada la relevancia y la utilización generalizada de los circuitos electrónicos, su comprobación cobra especial importancia. Además, para mantener una relación coste/prestaciones favorable es preciso que tal

comprobación, el test del circuito, se haga de manera eficiente. Sin embargo, con las tecnologías de integración LSI y VLSI el test de los circuitos presenta serios problemas y se ha hecho más difícil. Esta situación se debe al incremento de la **complejidad** de los circuitos a testear y a la posible **inaccesibilidad** de los diferentes subcircuitos incluidos en los chips LSI y VLSI. Como consecuencia, el test requiere herramientas de CAD cada vez más sofisticadas y equipos de test cada vez más precisos y esta tendencia se mantiene puesto que las tecnologías de circuitos integrados avanzan hacia circuitos más veloces y con mayores densidades de integración. Cabe destacar el hecho de que ha habido circuitos que se han diseñado y construido pero que no se han comercializado porque su comprobación era imposible o los hacía demasiado costosos.

Básicamente, la comprobación de un circuito se realiza a partir de la comparación de las respuestas del mismo ante una serie de estímulos, con las que se obtendrían en el circuito correcto. Esto implica que hay que determinar el conjunto de estímulos, llamados **patrones de test**, y acceder al circuito para aplicarlos. La selección del conjunto de patrones de test para comprobar la presencia de defectos en los circuitos lógicos es un problema NP-completo, es decir, cualquier algoritmo general para encontrar la solución óptima a este problema tiene complejidad exponencial. Así pues, en los circuitos LSI/VLSI los procedimientos usuales de búsqueda de patrones significan un tiempo de cálculo y unas necesidades de memoria demasiado elevadas.

De la misma forma que el problema de la selección de patrones de test en circuitos lógicos, existen otros problemas centrales del diseño lógico que son NP-completos. Como ejemplo se puede citar la minimización de funciones booleanas y la asignación de estados en máquinas secuenciales. Esta dificultad ha hecho que aumentara la demanda de técnicas de diseño analíticas y utilizables para circuitos complejos. Es una situación análoga a la que plantea el diseño de sistemas de control óptimos: el sistema se linealiza y después se diseña un sistema aproximado de control lineal óptimo utilizando técnicas basadas en la transformada Z, en la de Laplace o en la de Fourier. Siguiendo esta idea, se han utilizado técnicas espectrales, basadas en transformadas como la de Walsh y Chrestenson fundamentalmente, para el análisis, diseño y test de circuitos digitales [KAR85, LEC71, 81, MIL81, 85, MOR85, TRA85]. Estas transformadas, por otra parte, se pueden considerar generalizaciones de las transformaciones clásicas para el tratamiento de los circuitos digitales. Así, las técnicas de transformación en el dominio espectral se han utilizado en el análisis y clasificación de funciones

booleanas [HUR85, LEC71], diseño de circuitos digitales, y diagnóstico de fallos en circuitos digitales descritos tanto a nivel de puerta como a nivel de transistor [ERI86, HSI86, HUR85, KAR81, MÍL, RUI89, SUS83, DAR89]. Parten de un procedimiento de transformación rápida para obtener la descripción de la función en el dominio espectral, donde se realiza el análisis, la clasificación y el estudio de los cambios que originan los defectos del circuito. La transformada que se ha utilizado con más frecuencia ha sido la de Walsh, o su generalización para circuitos multivaluados, la transformada de Chrestenson [MOR85]. La presente memoria desarrolla una nueva transformación espectral, a la que se ha denominado Transformada de Tamari, puesto que él fue quien sugirió la base a partir de la que se construye [TAM52]. Se ha elaborado un algoritmo de cálculo rápido para la misma y se ha estudiado su utilización en la síntesis y en el test de circuitos lógicos.

El que los problemas que plantea el test de los circuitos lógicos sea un problema NP-Completo constituye un límite teórico que predice el fracaso de todo intento de buscar un algoritmo o una técnica que sea aplicable con efectividad razonable a todo tipo de circuitos. La solución al problema del test de un circuito concreto pasa por la selección de la técnica apropiada entre las que se han descrito en la bibliografía. Esta es la tesis fundamental de la presente memoria. Como consecuencia, no se trata de mostrar la técnica para realizar el test de los circuitos lógicos, sino de presentar una técnica más, analizar su utilidad exponiendo sus ventajas e inconvenientes para así delimitar las características del tipo de circuitos que constituyen su marco de aplicación. De esta forma, antes de pasar a describir las técnicas que se han desarrollado en la presente memoria, se ha creído conveniente tratar con detalle el test de los circuitos lógicos VLSI, los problemas que plantea y las soluciones que se han propuesto en la literatura. A ello se ha dedicado el capítulo 1. Se ha buscado en ese capítulo exponer, de forma autosuficiente, las cuestiones relativas al test, de ahí su longitud.

En los capítulos 2, 3 y 4 se describe la transformada de Tamari, y se estudia su utilización en la síntesis y en el test de circuitos lógicos multivaluados y, fundamentalmente, binarios.

El capítulo 5 se dedica a considerar los sistemas tolerantes a fallos, se introduce la terminología al uso y se explica la necesidad de estos sistemas. Uno de los presupuestos de un sistema tolerante a fallos es que se pueda conocer cuándo un módulo falla, para sustituirlo por otro. También se consideran en ese capítulo las posibles soluciones, una de las cuales la constituye el test concurrente de los módulos. En los capítulos 6 y 7 se

describe un procedimiento que hace uso de los coeficientes espectrales de Tamari para diseñar un circuito testeable concurrentemente.

Finalmente, el capítulo 8 recoge las conclusiones y resume las principales aportaciones de la presente memoria, así como las líneas a seguir en investigaciones futuras.

Para facilitar la lectura de la presente memoria conviene tener en cuenta las siguientes convenciones:

- Cada capítulo se ha dividido en secciones, las cuales a veces incluyen uno o más apartados. Cuando se hace alusión a cualquier parte del texto se indica el número del capítulo seguido del de la sección y, en su caso, del número de apartado. Ejemplos: sección 1.4, apartado 1.4.2.
- Las figuras y las tablas utilizadas están numeradas por capítulos. Ejemplos: figura 1.20, tabla 1.2.
- Las expresiones matemáticas que son referenciadas tienen igualmente una numeración por capítulos. Las referencias a éstas se indican mediante sus números entre paréntesis. Ejemplo: (6.18)
- Las referencias bibliográficas se indican en el texto mediante las tres primeras letras del apellido del autor seguidas por el año de publicación del trabajo. Si hubiera varios trabajos con la misma denominación se añade una letra. Ejemplos: [ABR82], [ABR82, 83b].

INDICE

INTRODUCCION.	1
INDICE	5
CAPITULO 1. EL TEST DE CIRCUITOS INTEGRADOS VLSI.	9
1.1 Problemas en el test de circuitos integrados.	12
1.2 Las soluciones al problema del tiempo de test.	15
1.2.1 División del circuito en subcircuitos con menos entradas.	16
1.2.2 Selección de patrones de test.	17
1.2.2.1 Selección determinista de patrones.	17
1.2.2.2 Selección pseudoaleatoria de patrones de test.	18
1.2.3 Tipos de faltas.	19
1.2.4 Evaluación del test.	25
1.2.5 Herramientas para la generación automática de patrones.	29
1.2.6 Estimación de la longitud del test pseudoaleatorio.	30
1.3 Las soluciones al problema de la accesibilidad de las entradas.	33
1.4 El problema del análisis de las respuestas.	36
1.4.1 Evaluación del enmascaramiento	37
1.5 Las soluciones al problema de la observabilidad de las salidas.	39
1.6 Las soluciones para el test de circuitos secuenciales.	40
1.7 Las soluciones al problema de los defectos que cambian la naturaleza combinatorial del circuito.	40
1.8 Las soluciones a los defectos temporales.	41
1.9 Las soluciones al problema de la degradación.	41
1.10 Clasificación de los procedimientos de test de circuitos integrados.	42
1.11 Objetivos y contenidos de la presente memoria.	44

CAPITULO 2. LA TRANSFORMADA DE TAMARI.	47
2.1 La Transformada de Tamari.	49
2.2 Un algoritmo rápido para la Transformada de Tamari (FTT).	53
2.2.1 Síntesis utilizando la Transformada de Tamari.	56
2.2.2 Representación gráfica del cálculo de la FTT.	57
2.2.3 Complejidad de cálculo de la FTT.	60
2.3 Obtención aislada de los coeficientes espectrales de Tamari.	61
2.4 El Test de circuitos utilizando la FTT.	63
2.4.1 Transformación de la FTT con las faltas de anclaje.	66
2.4.2 Coeficientes espectrales para el test de faltas de anclaje simples.	69
2.5 Test de faltas de anclaje en circuitos producto/suma módulo-p.	71
2.6 Signatura espectral en redes de puertas suma módulo-p.	72
2.7 Obtención de la signatura en redes sin reconvergencias.	73
2.8 Procedimiento espectral de generación de patrones de test.	73
2.9 Conclusión.	74
CAPITULO 3. LA TRANSFORMADA DE TAMARI EN CIRCUITOS BINARIOS.	75
3.1 La Transformada de Tamari de funciones binarias.	76
3.1.1 Propiedades de los coeficientes espectrales.	78
3.1.2 Obtención secuencial de los coeficientes espectrales	80
3.2 Los coeficientes espectrales en el test de circuitos binarios	81
3.2.1 Detección de las faltas de anclaje.	82
3.2.2 Detección de las faltas de cortocircuito en las entradas.	84
3.2.3 Detección de faltas en circuitos AND-EXOR.	86
3.2.4 Detección de faltas en circuitos de paridad.	87
3.2.5 Detección de faltas en circuitos sin reconvergencias.	88
3.2.6 Detección de faltas en PLAs AND-OR.	88
3.3 Procedimiento de selección de patrones de test para las faltas de anclaje	92
3.4 Conclusión	98

CAPITULO 4. PROCEDIMIENTO MODULAR PARA EL TEST ESPECTRAL.	99
4.1 Procedimiento modular para el cálculo de la transformada de Tamari.	100
4.2 Cambios en los coeficientes espectrales	105
4.3 Procedimiento modular para la selección de patrones de test	106
4.4 Procedimiento para circuitos con interconexión unidimensional de módulos	108
4.5 Ejemplo de aplicación del procedimiento modular	109
4.6 Conclusión	112
CAPITULO 5. EL TEST CONCURRENTE DE CIRCUITOS INTEGRADOS.	113
5.1 Un modelo del proceso de detección.	115
5.2 Los circuitos testeables concurrentemente.	121
5.2.1 Procedimientos Tipo I	121
5.2.2 Procedimientos Tipo II	123
5.3 Evaluación de los procedimientos de test concurrente.	124
5.3.1 Estimación del tiempo de latencia (TL).	125
5.4 El procedimiento de test concurrente propuesto.	127
5.4.1 Análisis y síntesis de circuitos testeables concurrentemente.	130
5.5 Conclusión.	132
CAPITULO 6. DISEÑO DEL MODULO EXTRACTOR.	133
6.1 El módulo extractor como compresor de salidas.	134
6.2 Efectividad del extractor en términos de los coeficientes de Tamari.	134
6.3 Modelos de error.	139
6.3.1 Obtención experimental de un modelo de error.	141
6.4 Procedimiento basado en el Recocido Simulado	145
6.5 Conclusión.	149

CAPITULO 7 .DISEÑO DE LOS MODULO PREDICTOR Y MONITOR DE PATRONES.	151
7.1 Selección de los patrones de test.	152
7.1.1 Selección determinista.	155
7.1.2 Selección aleatoria.	156
7.2 Procedimiento de generación de test y síntesis del módulo monitor.	160
7.2.1 Algoritmo de selección aleatoria de patrones.	168
7.3 Síntesis del circuito predictor.	170
7.3.1 Ejemplo de diseño para el test concurrente	172
7.4 Conclusión.	176
CAPITULO 8. PRINCIPALES APORTACIONES Y CONSIDERACIONES FINALES.	177
APENDICE I. EVALUACION DEL ENMASCARAMIENTO EN UN LFSR.	181
APENDICE II. PROCEDIMIENTO DE SINTESIS AND-EXOR MULTIFUNCIONAL.	185
APENDICE III. PROPIEDADES DE LOS COEFICIENTES ESPECTRALES BINARIOS	197
APENDICE IV. TEST DE FALTAS DE ANCLAJE DE TRANSISTOR EN CIRCUITOS CMOS	199
APENDICE V. SINTESIS DEL MODULO EXTRACTOR MEDIANTE UNA RED NEURONAL DE HOPFIELD.	203
APENDICE VI. LISTADOS DE DESCRIPCION DE RED Y DE SCOAP PARA LA ALU 74181	209
REFERENCIAS.	219

CAPITULO 1. EL TEST DE CIRCUITOS INTEGRADOS VLSI.

El desarrollo tecnológico ha hecho posible la inclusión de miles de componentes electrónicos y sus interconexiones en un sólo circuito integrado: se trata de los circuitos de muy alta escala de integración o circuitos VLSI (Very Large Scale Integration). Con el uso de estos circuitos se consigue reducir el coste y el tamaño de los sistemas electrónicos y aumentar sus prestaciones y fiabilidad [WIL83, TSU87]. Sin embargo, para que eso sea efectivamente cierto es necesario que los circuitos VLSI se puedan testear económicamente [SET85a].

Testear un circuito integrado (CI) es comprobar su correcto funcionamiento una vez fabricado [AGR88]. En la figura 1.1 se muestran las distintas etapas por las que pasa un circuito VLSI, desde que se establecen las especificaciones que ha de satisfacer hasta que, finalmente, está disponible. El test se ha de llevar a cabo una vez se han fabricado los circuitos y permite determinar cuales son aceptables.

Los aspectos que encierra la comprobación de un circuito integrado, tal como se indica en la figura 1.2 [TSU87], son:

- a) Comprobar que se alcanzan los valores de tensión e intensidad previstos y que el comportamiento dinámico se ajusta a las especificaciones (test paramétrico).
- b) Comprobar el comportamiento lógico (test lógico).

En la comprobación de un circuito hay que distinguir entre test y verificación: el test se refiere a la comprobación una vez fabricado el

circuito, mientras que la verificación consiste en comprobar las especificaciones del circuito durante su síntesis. Como se ve en la figura 1.2, la verificación se realiza durante la etapa de diseño mediante el uso de simuladores adecuados para cada uno de los niveles de descripción utilizados y mediante los propiamente llamados verificadores. Por otra parte, la figura 1.2 ilustra el hecho de que el test no sólo lo aplica el fabricante del circuito una vez manufacturado éste, sino que el test se lleva a cabo también en otros momentos de la vida del circuito integrado [AGR88]. Así, además del test que realiza el fabricante, **test de fabricación o de prototipos**, el usuario debe realizar el test del circuito que ha adquirido para asegurarse de que es aceptable, **test de entrada**, y una vez se han montado los circuitos integrados se ha de comprobar el correcto funcionamiento de las placas, **test de placas**, y del sistema completo, **test del sistema**. Por último, durante la vida activa del sistema ha de llevarse a cabo una serie de tests periódicos, o **tests de mantenimiento**, para asegurar que el sistema funciona correctamente.

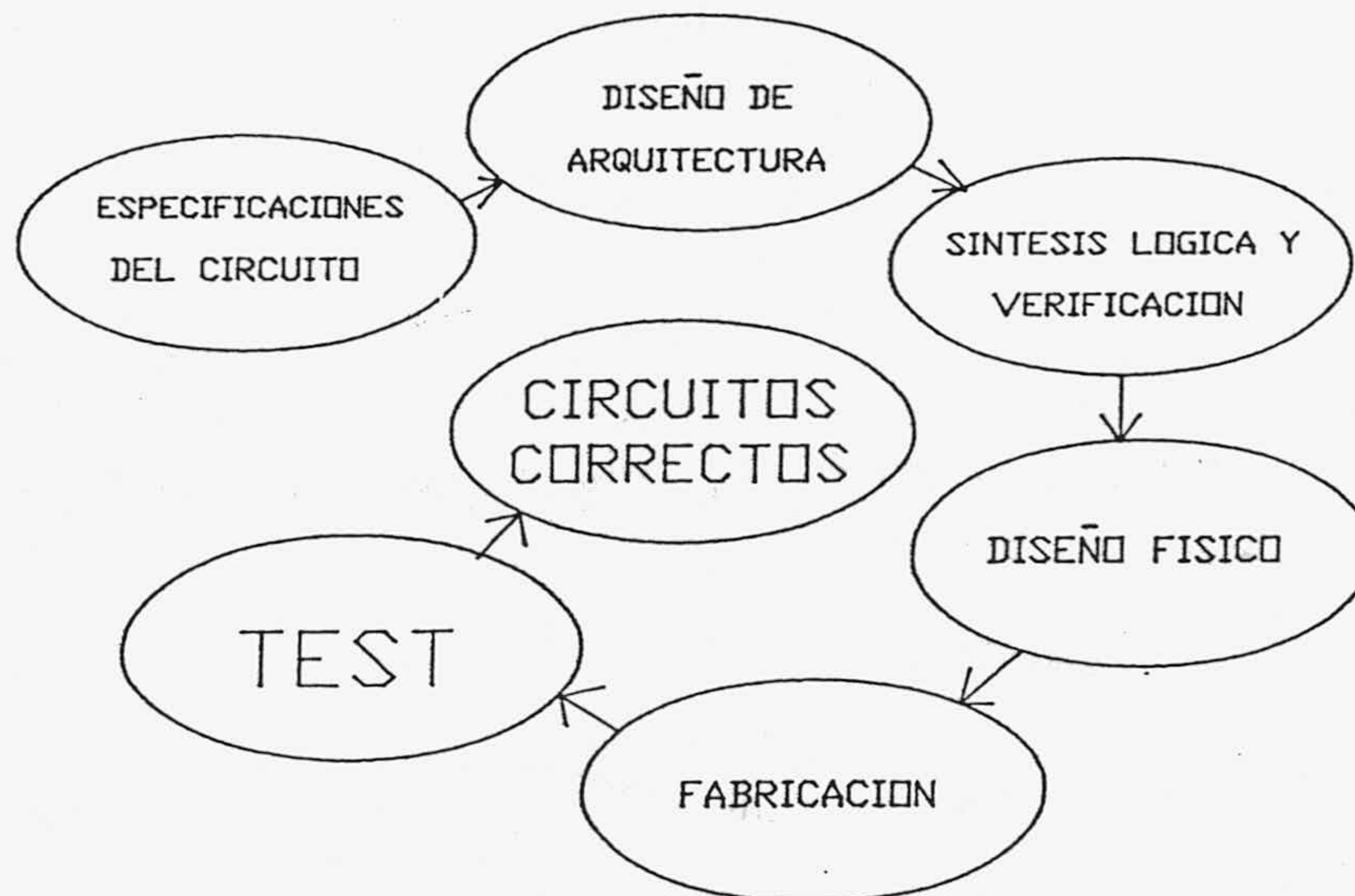


Figura 1.1. Etapas en el desarrollo de un circuito VLSI.

Para aplicar el test una vez fabricado el circuito, se monta éste en un **equipo automático de test (ATE)** y se aplica una serie de estímulos a los terminales de entrada del circuito, comparando las respuestas obtenidas con las esperadas. Según esto, es necesario [AGR88] **generar las combinaciones de entradas a aplicar, conocidas como patrones de test, y evaluar su efectividad**. Con estos patrones se programa el equipo de test, ATE, teniendo en cuenta la forma en que han de aplicarse para realizar el test paramétrico y el test lógico según se ha establecido en la **planificación del test**. Así pues, la figura 1.1 se puede completar con la figura 1.3, que muestra cómo se insertan

las actividades relacionadas con el test dentro del proceso de diseño y fabricación del circuito. Por consiguiente, el test de un circuito integrado implica tener presente [RAS82]:

- a) El equipo de test de que se disponga.
- b) El software para generar el test, evaluar su efectividad y analizar las respuestas.
- c) El diseño del circuito a testear, que debe hacerse de forma que permita y facilite el test.

VIDA DEL CIRCUITO	DISEÑO	Simuladores Verificadores de Coherencia Verificadores de Reglas Extractores de Parámetros Simulación post-layout	Funcionales Lógicos Temporales Eléctricos
	FABRICACION	Test Paramétrico Test de Obleas Test de Encapsulado	
	Test de prototipos Test de Entrada Test de Placas Tests Periódicos (VIDA ACTIVA)		

Figura 1.2. Tests aplicados durante la vida activa de un circuito integrado.

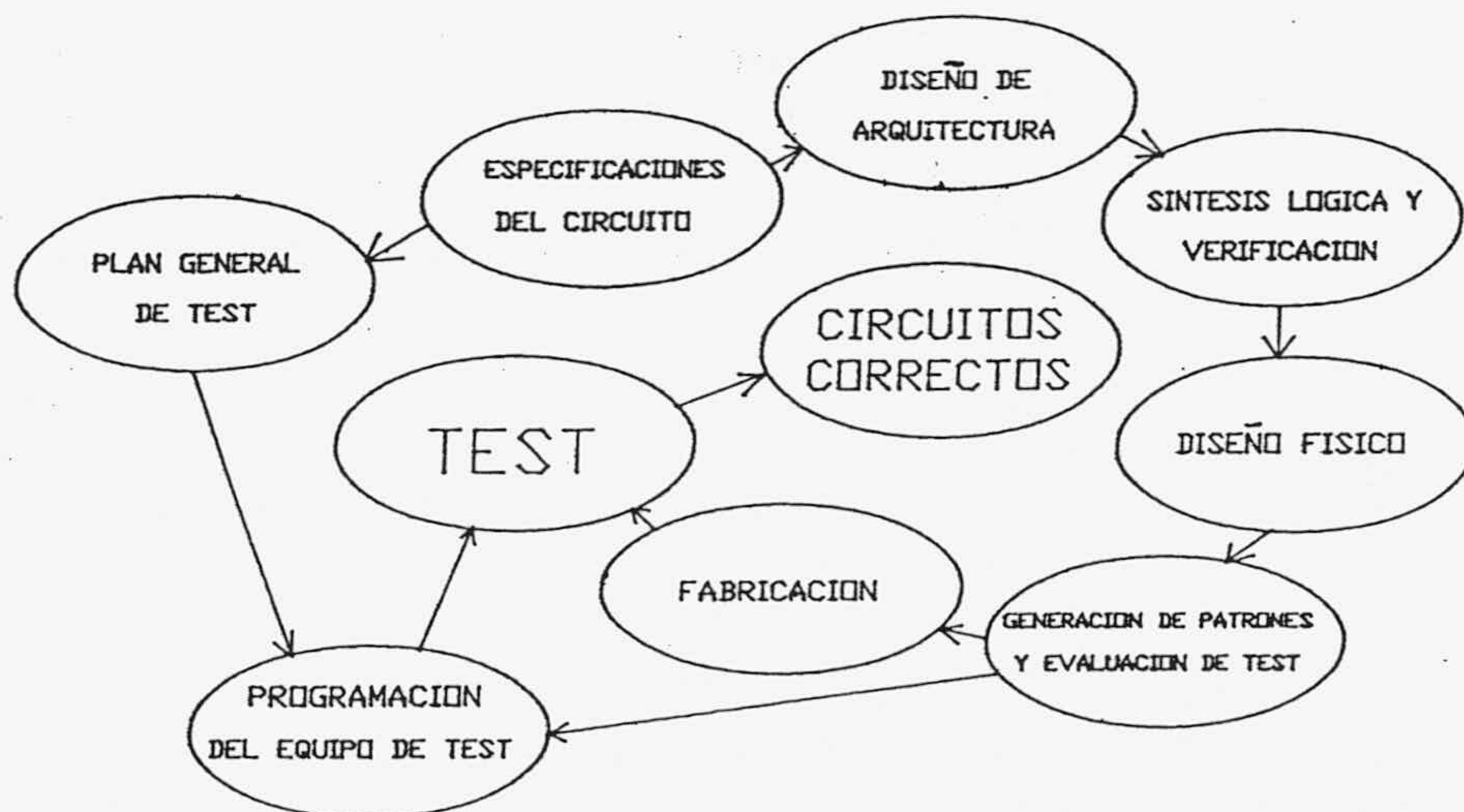


Figura 1.3. Actividades relacionadas con el test en el desarrollo de un circuito integrado.



En esta memoria se considera el test de circuitos integrados digitales desde el aspecto de verificación de su comportamiento lógico: a partir del desarrollo de las tecnologías de integración MSI y LSI y con la disminución del número de elementos de circuito accesibles, el test paramétrico del interior del circuito queda garantizado a través del test lógico [RAS82]. El test paramétrico se limita a los elementos periféricos del circuito integrado a los que se tiene acceso. Este primer capítulo se dedica a ofrecer una panorámica de los problemas que presenta, de las soluciones que se han planteado y de los distintos procedimientos de test a los que esas soluciones han dado lugar. El objetivo es el de introducir la terminología al uso en este campo, establecer una taxonomía que establezca un cierto orden en la diversidad de procedimientos existentes y plantear los fines del presente trabajo a partir de las cuestiones mostradas.

1.1. Problemas en el test de circuitos integrados.

Testear un circuito lógico no parece tener mucho interés desde el punto de vista conceptual: bastaría aplicar todas las posibles combinaciones de entradas y comprobar si las salidas obtenidas corresponden a las que cabe esperar según el diseño del circuito.

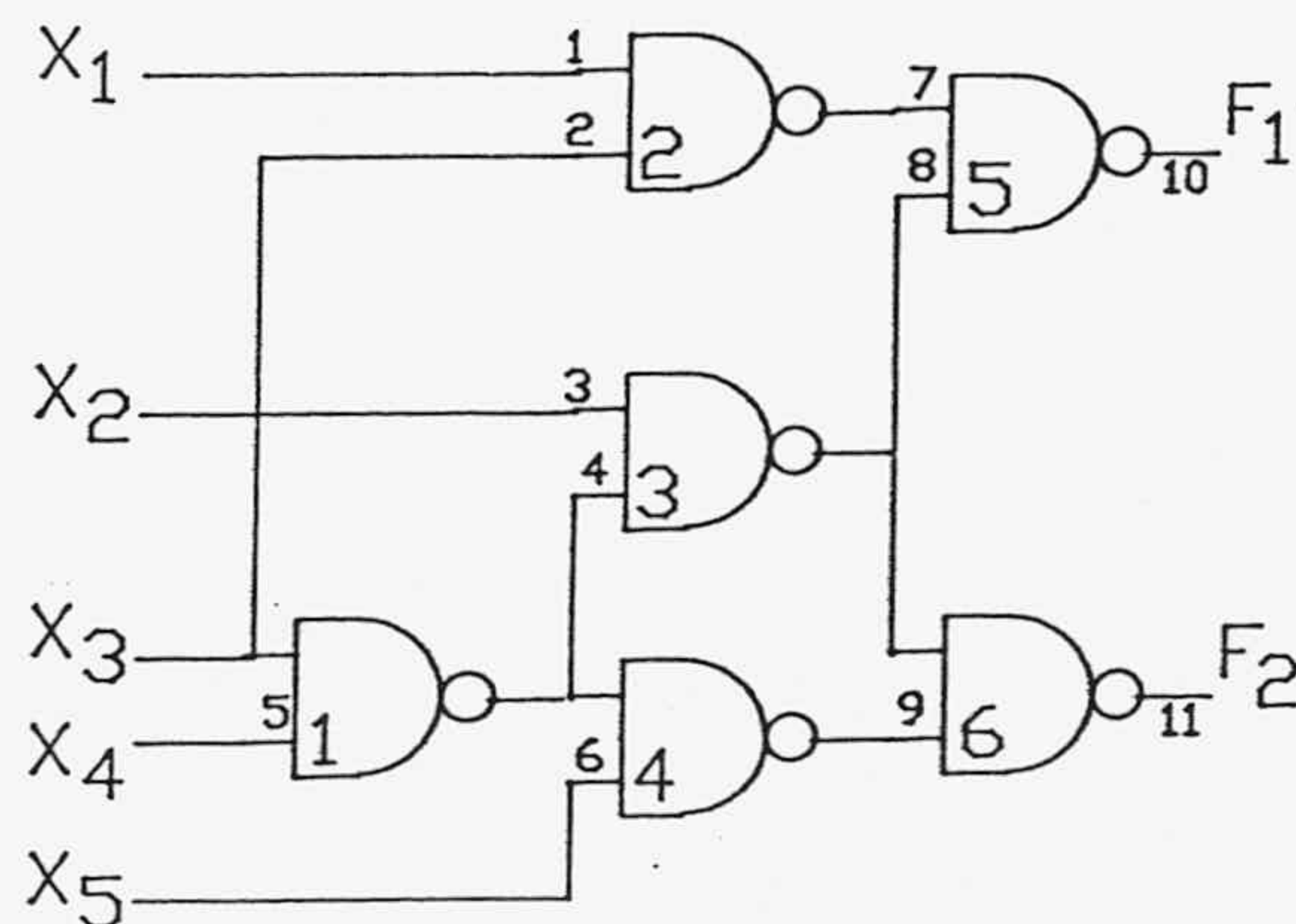


Figura 1.4. Ejemplo de circuito lógico.

Sea el ejemplo de la figura 1.4. Se trata de un circuito con 5 entradas que sintetiza dos funciones, F_1 y F_2 , utilizando 6 puertas NAND de dos entradas:

$$F_1(x_1, x_2, x_3, x_4) = x_1x_3 + x_2x_3' + x_2x_4'$$

$$F_2(x_2, x_3, x_4, x_5) = (x_3x_4)'(x_2 + x_5)$$

(1.1)

Se puede comprobar su funcionamiento aplicando las 2^5 combinaciones de entradas y comparando las salidas que se van obteniendo con sus valores correspondientes al circuito sin defectos, que se calculan a partir de las expresiones lógicas de las funciones sintetizadas (1.1). Así, para las entradas $x_1=1, x_2=0, x_3=1, x_4=0, x_5=1$, las salidas que se han de obtener en el circuito correcto son $F_1=1, F_2=1$. Si se detecta una variación en alguno de estos valores se dice que se ha producido un error. Este se representa en un circuito digital binario mediante un vector (e_1, e_2, \dots, e_N) , donde N es el número de salidas del circuito ($N=2$ en el ejemplo) y $e_j=1$ si el valor que toma la salida F_j no es el correcto ($e_j=0$ en caso contrario).

Sin embargo, este esquema tan simple presenta una serie de problemas que lo hacen inviable en la mayoría de los casos. Estos están relacionados con:

A. El tiempo de test: si el número de entradas del circuito es elevado puede llevar un tiempo considerable aplicarlas todas. Por ejemplo, si suponemos un circuito con unas 20 entradas y 40 biestables accesibles exteriormente, el número de patrones a aplicar sería 2^{60} (2^{20+40}) y el tiempo necesario, a razón de 10^6 patrones/sg., sería del orden de 35000 años.

B. La accesibilidad a las entradas: no siempre es posible acceder a las entradas de un circuito y no es inmediato, en la mayoría de los casos, conseguir que el estado de los biestables de un circuito sea el deseado.

C. El análisis de respuestas: es preciso tener algún procedimiento que permita recordar la respuesta correcta a cada una de las entradas aplicadas para compararla con la que se obtiene. Si se opta por almacenar esas respuestas, se necesitará una cierta cantidad de memoria que puede ser excesiva. Como en el caso del circuito de 20 entradas y 40 biestables considerado más arriba: si el circuito tiene N salidas, haría falta una memoria de $N2^{60}$ bits, es decir $N2^{20}$ Gigabits (!).

D. La observabilidad de las salidas: es preciso poder acceder a las salidas del circuito para captar las respuestas a cada combinación de entradas.

E. El test de circuitos secuenciales: en un circuito secuencial el test exhaustivo no sólo significa aplicar todas las posibles combinaciones de entradas sino también todas las combinaciones de estados. Ahora bien, en muchos casos resulta difícil hacer que un circuito secuencial alcance un determinado valor en sus variables de estado: hay que estar seguro del estado actual y determinar una secuencia de entradas que lo lleven al estado deseado.

F. Los defectos que hacen secuencial un circuito combinacional: ciertos defectos modifican la naturaleza combinacional de un circuito y no se detectan mediante un único patrón sino a través de una cierta secuencia de patrones.

Considérese la figura 1.5; para detectar si hay un corte en T1, hay que asegurarse que Z está en alto cuando se aplica el patrón (A,B)=(0,1).

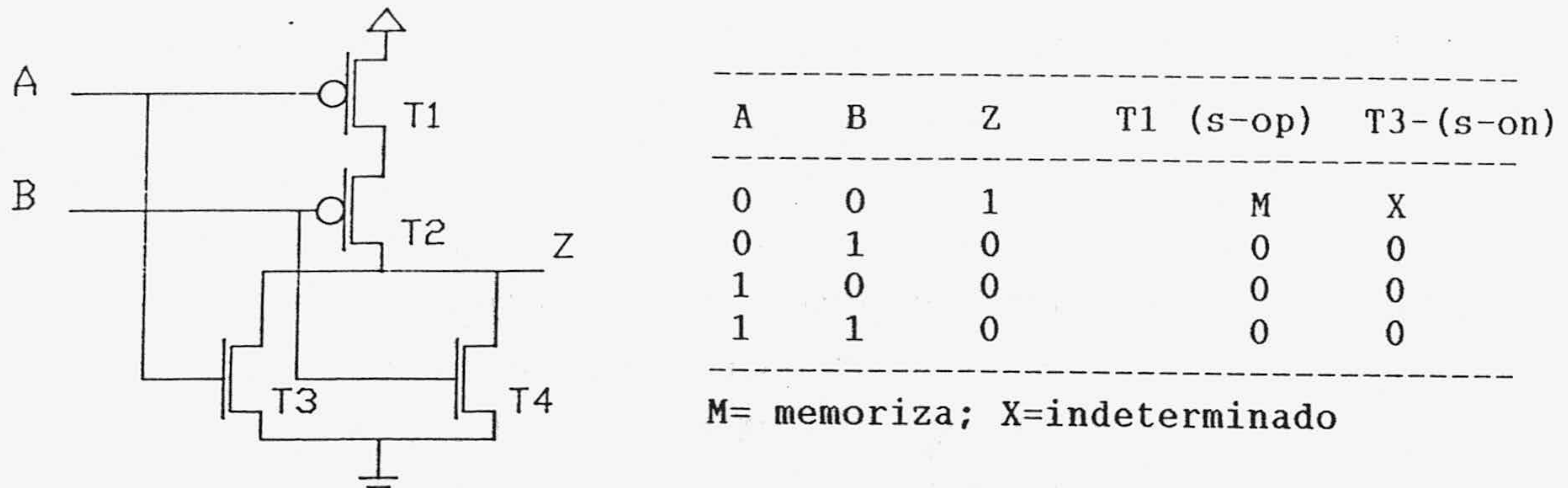


Figura 1.5. Puerta NOR CMOS.

G. Los defectos temporales: aunque el circuito haya sido comprobado frente a defectos de fabricación como pueden ser los contactos incorrectamente realizados, los defectos en el óxido, la electromigración o la corrosión [ABR86b, BUR88, EAS77, FAN85, MAN84, SHE85], durante su funcionamiento puede verse afectado por defectos que hagan que se comporte de forma anómala sólo durante ciertos intervalos de tiempo. Esos defectos temporales [LAL85] no se detectarán a no ser que la aplicación del test coincida con su manifestación. Entre las causas de estos defectos se encuentran la influencia del entorno a través de radiaciones cósmicas o partículas α , la electromigración y la inyección de electrones calientes [ABR86b, GHA82].

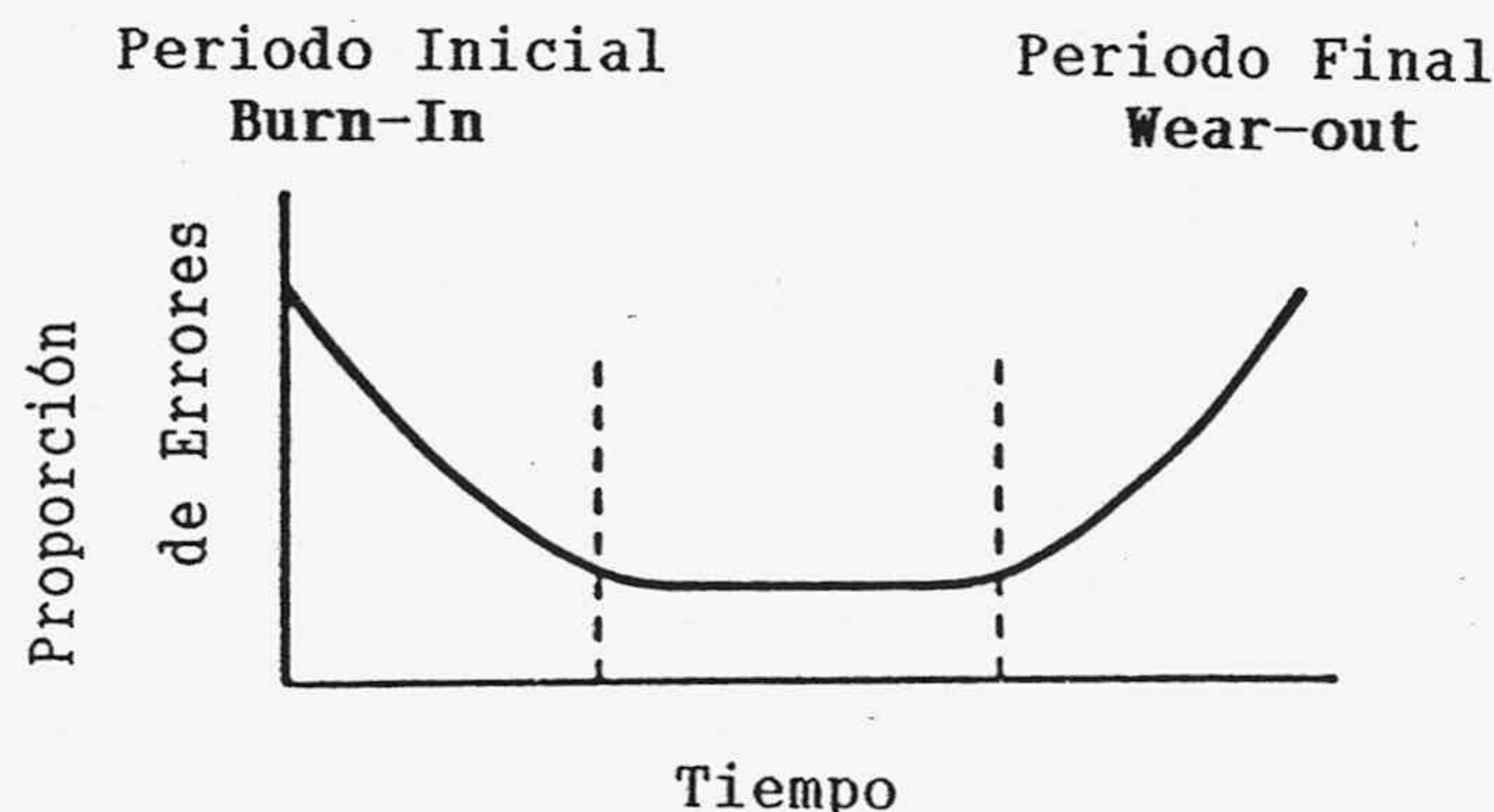


Figura 1.6. Variación de la proporción de errores en el tiempo.

H. La degradación de los circuitos: puede ocurrir que el uso prolongado del circuito lo deteriore de manera que deje de funcionar correctamente. En la figura 1.6 se muestra la proporción de errores en el comportamiento de los

componentes electrónicos (fallos) en función del tiempo de utilización [LAL85]. Se observa un periodo inicial, periodo de **burn-in**, con un elevado número de fallos debidos a componentes defectuosos que no fueron correctamente testeados en su momento, y un periodo final, periodo de **wear out**, en el que la proporción de fallos también aumenta debido a mecanismos de envejecimiento como, por ejemplo, la electromigración, la inyección de electrones calientes y la pérdida de carga por difusión [ABR86b, REI88].

Todos estos problemas se presentan en los circuitos digitales integrados y se ven agravados a medida que la escala de integración y la complejidad de los sistemas electrónicos aumenta. Así, las tarjetas de los sistemas electrónicos incluyen una gran cantidad de estos circuitos integrados, siendo el acceso a sus entradas y salidas muy difícil, y no digamos el acceso a los módulos que contiene cada circuito integrado [ABA83b]. En lo que a los fallos temporales se refiere, estos son más probables en los circuitos VLSI, ya que las geometrías y los márgenes de ruido se ven reducidos [ABR86b]. Por otra parte, al abaratare el costo de los circuitos integrados VLSI se pueden desarrollar sistemas en los que se utilice una gran cantidad de circuitos iguales para aprovechar el paralelismo. Para que estos sistemas presenten cierta fiabilidad es imprescindible dotarles de algún procedimiento que permita que se vayan testeando a la vez que funcionan [LAL85, TEW89].

Se ha planteado una serie de soluciones a estos problemas que han dado lugar a diversos procedimientos aplicables con mayor o menor efectividad a cada circuito integrado [RAS82]: según los problemas que un procedimiento de test resuelva será adecuado o no para comprobar un circuito concreto en función de dónde radiquen las dificultades para testearlo. Esta es una de las decisiones que hay que tomar en la fase de planificación de test (figura 1.3).

En las secciones 1.2 a 1.9, se describen las diversas maneras de encarar los problemas del test de CI y las cuestiones que plantean. Posteriormente, en la sección 1.10, se establece una clasificación de los diversos procedimientos de test propuestos a partir de las soluciones descritas, indicándose qué problemas del test permiten eludir con mayor eficacia.

1.2. Las soluciones al problema del tiempo de test.

Para disminuir el tiempo de test se han buscado procedimientos que reduzcan el número de patrones de test necesarios. Entre las posibilidades que se barajan están la división del circuito en varios subcircuitos con menos

entradas, para que cada subcircuito se pueda testear de forma exhaustiva, la búsqueda de patrones para detectar cada uno de los defectos y, por último, la selección de patrones de forma aleatoria sin tener en cuenta qué defectos permite detectar cada uno de ellos. Se consideran en las subsecciones 1.2.1 y 1.2.2. El resto de las subsecciones se dedican a exponer las cuestiones más importantes que implica la selección de patrones, determinista o aleatoria. Concretamente, en la sección 1.2.3 se considera el modelado de defectos mediante faltas, en la 1.2.4 la evaluación del conjunto de patrones seleccionado para testear un circuito, en la 1.2.5 se consideran las herramientas disponibles para la generación de patrones de test y finalmente, la subsección 1.2.6 se dedica al problema de la determinación del número de patrones de test que deben tomarse en el caso de selección aleatoria.

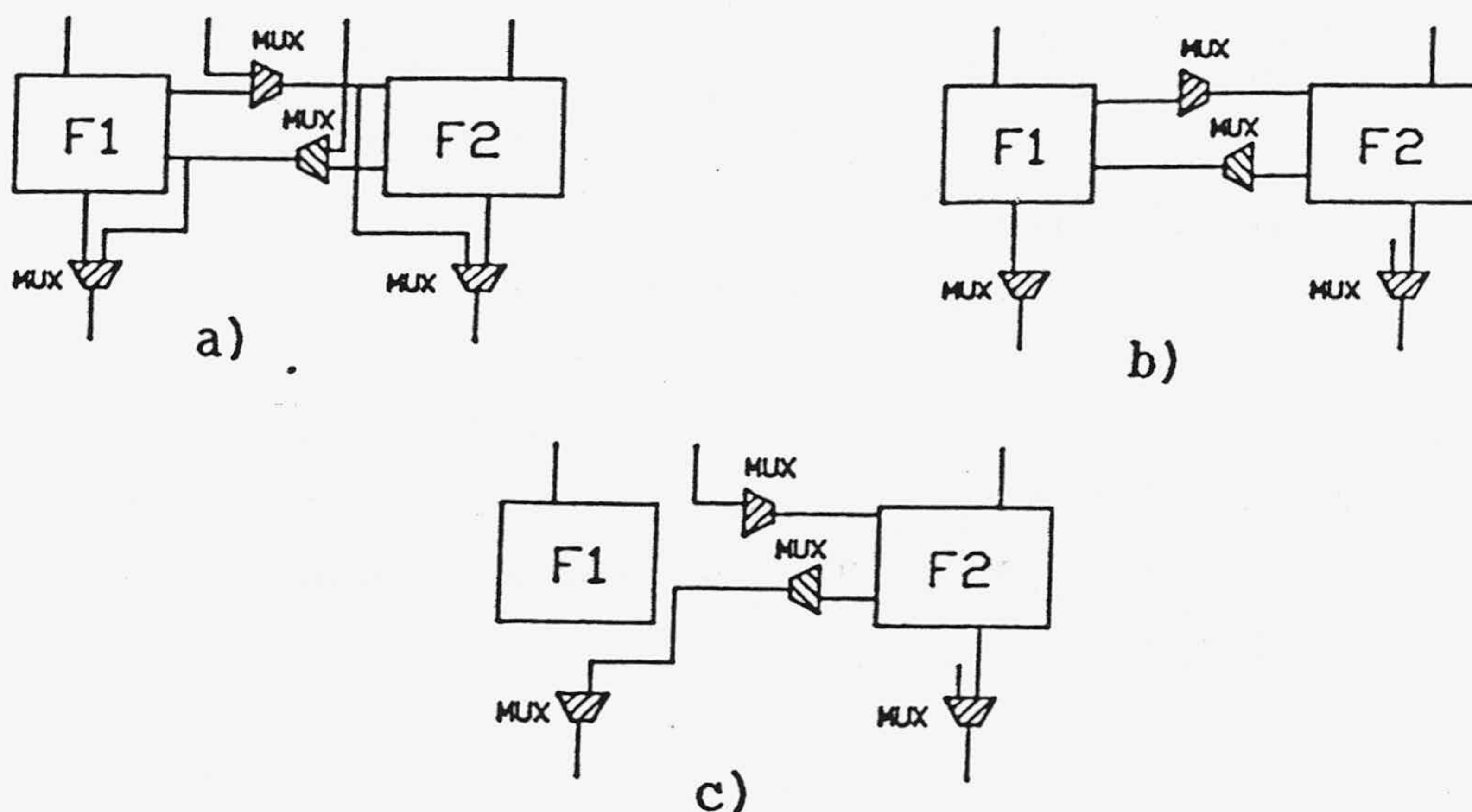


Figura 1.7. Circuito con dos módulos (F1 y F2). (a) Modificaciones para el test pseudoexhaustivo; (b) funcionamiento normal; (c) test del módulo F2.

1.2.1. División del circuito en subcircuitos con menos entradas.

Esta solución se puede adoptar si es posible distinguir en el circuito a testear varios subcircuitos que no dependan de todas las entradas y que sean independientes entre sí. De esta manera todos los subcircuitos se pueden testear en paralelo y, al tener menos entradas, el número de combinaciones necesaria para el test exhaustivo de cada uno es menor [MCL84, GOL88]. Así, si un circuito con N entradas se puede dividir en dos circuitos de $N/2$ entradas cada uno, el número de patrones necesarios para el test exhaustivo pasa de 2^N a $2^{N/2}$.

En el ejemplo de la figura 1.4, puesto que F1 sólo depende de x_1 , x_2 , x_3 y x_4 y F2 de x_2 , x_3 , x_4 y x_5 , si se aplica $x_1=x_5$ con todas las combinaciones de x_1 , x_2 , x_3 y x_4 , se estará realizando un test exhaustivo de cada una de las funciones, pero se necesitan 2^4 patrones frente a las 2^5 posibles combinaciones de entradas.

A veces, para poder utilizar esta solución es necesario realizar ciertas modificaciones en el circuito para que se pueda dividir en partes independientes en el momento de aplicar el test [ROB84]. En la figura 1.7 se muestra un ejemplo de esta posibilidad para un circuito con dos bloques, F1 y F2, al que se le han añadido multiplexores para poder separar ambos bloques.

1.2.2. Selección de patrones de test.

Una combinación de entradas permite detectar un defecto si lo pone de manifiesto ocasionando un error en alguna de las salidas del circuito. En ese caso tal combinación de entradas se podría utilizar como **patrón de test** para tal defecto. Según la forma de seleccionar los patrones de test se puede distinguir entre procedimientos deterministas (apartado 1.2.2.1) y procedimientos pseudoaleatorios (apartado 1.2.2.2).

1.2.2.1. Selección determinista de patrones de test.

Estos procedimientos pretenden la determinación de un conjunto mínimo de patrones de test buscando las combinaciones de entradas que, para cada defecto, ocasionan error en las salidas [ARM66, BOT77, ROT67, TIM82b]. Para predecir si se producirá error cuando se aplique una determinada combinación de entradas a un circuito defectuoso es necesario modelar los defectos físicos de los circuitos. Tales modelos reciben el nombre de **faltas**.

Supóngase el circuito de la figura 1.4. Considérese que determinado defecto del circuito origina que cuando a las entradas de la puerta NAND número 4 se aplica la combinación de entradas (1,1), su salida (nudo 9) no toma el valor lógico 1. El patrón de test para detectar este defecto debería hacer que los nudos 4 y 6 estuvieran a 1, y después poner de manifiesto el valor del nudo 9 en cualquiera de las salidas. Para que 4 esté a 1 basta hacer $x_3=0$ ó $x_4=0$, y con $x_5=1$ se consigue que 6 esté a 1. Por otra parte, para que el valor del nudo 9 se observe en F2 hay que hacer el nudo 8 igual a 1 y para ello (puesto que 4 está a 1) $x_2=0$. Así, para detectar el defecto que se está

considerando, se necesita un patrón de test en el que $x_2=0$, $x_3=0$ (ó $x_4=0$) y $x_5=1$. Como se ha podido comprobar a partir del proceso de obtención de este patrón de test, la facilidad para encontrar patrones depende de la observabilidad y de la controlabilidad del nudo en el que se manifieste el defecto a detectar [GOL79]: La controlabilidad de un nudo es una medida del número de combinaciones de entradas que permiten poner ese nudo a un valor determinado y la observabilidad es una medida del número de combinaciones de entradas que hacen que un determinado valor del nudo se ponga de manifiesto en alguna de las salidas del circuito. A partir de la controlabilidad y la observabilidad de cada uno de los nudos se han establecido diversas medidas de la facilidad con que se pueden testear los nudos de un circuito: es la testeabilidad de cada uno de esos nudos [AGR82, BEN84, GOL80]. Existen programas que permiten calcular las controlabilidades y observabilidades de los nudos del circuito y, consecuentemente, analizar sus testeabilidades. Entre estos, cabe destacar SCOAP [GOL80], CAMELOT [BEN84] y el algoritmo descrito en [SIN84] para circuitos MOS VLSI. Las distintas medidas de testeabilidad proporcionan información acerca de la implementación del circuito de cara a la facilidad con la que se pueden detectar los defectos que en él se produzcan: aunque no indican si un test va a detectar una falta dada, sí permiten estimar la probabilidad de detección de una falta [AGR82, SIN84].

Así pues, la selección determinista de patrones de test implica:

- a) buscar modelos de defectos adecuados y fáciles de aplicar, y
- b) disponer de procedimientos eficientes que automaticen el cálculo del patrón de test para cada falta y que sean aplicables a circuitos complejos.

Además, puesto que el número de defectos que puede presentar un circuito de complejidad media es muy elevado, se necesitan criterios para seleccionar las faltas a utilizar en la búsqueda de patrones de test [SHE85]. También sería de enorme interés conocer qué faltas tienen el mismo efecto, para así sólo considerar una de ellas (equivalencia de faltas o fault collapsing [SCH72]).

1.2.2.2. Selección pseudoaleatoria de patrones de test.

En estos procedimientos la determinación de los patrones de test se realiza tomando al azar un conjunto de combinaciones de entradas [SAS88,

SCH75, THE81, TIM83b]. El problema se plantea a la hora de asegurar que ese conjunto de patrones seleccionado aleatoriamente es capaz de detectar un número suficiente de los defectos del circuito. Como se verá detenidamente en el capítulo 7, para resolverlo también hay que recurrir a modelar los defectos mediante faltas y a seleccionar un conjunto de dichas faltas, ya sea para la búsqueda de expresiones que relacionen la probabilidad de detección de faltas con la longitud del test (el número de patrones que lo componen), o en la simulación del circuito con defectos para comprobar si alguno de los patrones seleccionados los pone de manifiesto a la salida.

Así pues, idear un procedimiento de selección de patrones de test eficiente significa enfrentarse con la necesidad de:

- a) **Modelar los defectos adecuadamente mediante faltas.**
- b) **Seleccionar un número suficiente de defectos para buscar los patrones o para la evaluación del test.**
- c.1) **Diseñar un procedimiento eficiente para buscar los patrones que detecten cada falta (procedimientos deterministas).**
- c.2) **Establecer expresiones que relacionen el número de patrones de test seleccionados con la capacidad de detección de defectos (procedimientos pseudoaleatorios).**

1.2.3. Tipos de faltas.

Como se ha indicado en la sección 1.2.2, para seleccionar el conjunto de patrones de test se necesita modelar los defectos del circuito mediante faltas. Una falta modela un defecto físico si permite la predicción de las salidas del circuito defectuoso para cada una de las entradas.

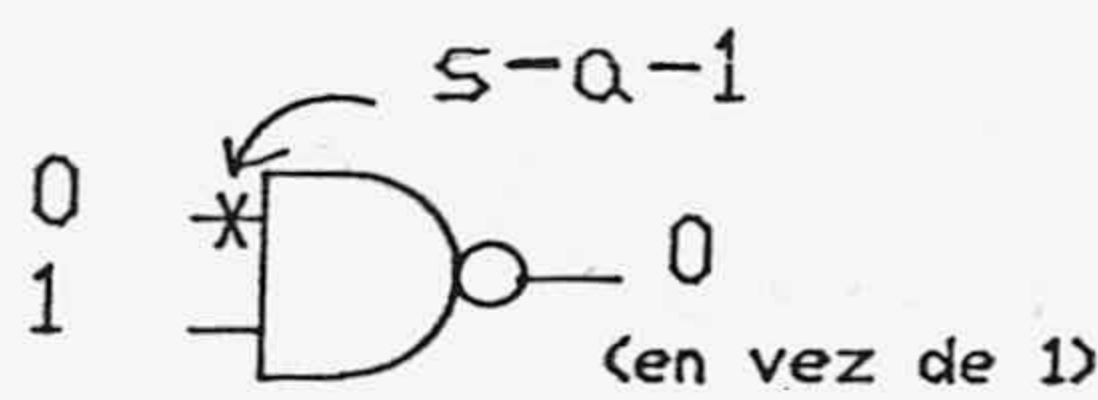
La forma de modelar un defecto depende del nivel de descripción del circuito que se va a utilizar en la generación del test y éste ha de ser lo suficientemente simple como para que la complejidad de los algoritmos de generación de patrones no sea excesiva [ABR86b]. Una descripción de los defectos que puede presentar un circuito VLSI se puede encontrar en [ABR86b, BUR88, EAS77, FAN85, MAN84, SHE85] y los distintos tipos de faltas que se han utilizado para modelarlos se pueden consultar, entre otros, en [ABR86b, MEI74, RED84, SCH72, TIM83, WAD78a, 78b]. Las faltas más frecuentemente utilizadas hasta el momento son las:

- de línea anclada (stuck-at).
- de cortocircuito (bridging)

- de contacto (crosspoint).
- de transistor anclado (stuck-on y stuck-open).

Los dos primeros tipos de faltas corresponden al nivel de descripción de puertas lógicas, en tanto que los dos últimos se utilizan con descripciones del circuito en el nivel de conmutador (switch-level) [ABR86b, HAY87, LIU87]. A continuación se va a describir cada una de ellas.

De todos esos modelos, el más comunmente usado ha sido el de línea anclada. Este modelo asume que cualquier defecto que se produzca en una puerta lógica se traduce en que alguna de sus entradas o su salida se mantiene fija al valor lógico 1 (falta de anclaje al 1 ó s-a-1) o al valor lógico 0 (falta de anclaje al 0 ó s-a-0). Así, si una puerta NAND tiene un defecto que se modela mediante una falta s-a-1 en una de sus entradas (figura 1.8), independientemente de los valores lógicos que se apliquen a esa entrada, actuará como si siempre estuviera en el valor lógico 1.



AB	C	A(s-1)
00	1	1
01	1	0
10	1	1
11	0	1

Figura 1.8. Anclaje a 1 (s-a-1) en una entrada de una puerta NAND.

Este tipo de falta constituye el modelo "clásico" sobre el que se han elaborado la mayoría de procedimientos automáticos de generación de patrones de test (ATPG), [AHR84, BES84, BRG85, FUN85]. Su aceptación se debe, fundamentalmente, a tres razones: es un modelo fácil de utilizar, ya que sólo implica mantener las líneas afectadas a un valor lógico constante; ofrece una buena representación de los principales defectos que se pueden presentar en las tecnologías TTL y MOS; y en la práctica se ha mostrado como un modelo que permite obtener tests efectivos (sección 1.2.4).

En la figura 1.9 se muestra la realización TTL de una puerta NAND. Los defectos 1 y 4 se pueden modelar como faltas s-a-1 en las entradas correspondientes (aunque en el caso del defecto 4 se puede ocasionar un segundo defecto, ya que la puerta anterior está sobrecargada); los defectos 2 y 3 se modelan como faltas de s-a-1 en la salida; y el defecto 5 se modela como una falta de s-a-0 en la entrada correspondiente.

La mayoría de los procedimientos de generación automática de patrones de test, además de suponer que todos los defectos que se puedan presentar en

un circuito se pueden modelar como faltas de anclaje, también consideran que, puesto que la probabilidad de que se presente más de un defecto en un circuito es pequeña, también lo es la probabilidad de que haya que utilizar más de una falta de anclaje, es decir, modelan el circuito defectuoso como un circuito con una falta de anclaje en alguna de sus líneas. Por consiguiente, esos procedimientos buscan un patrón de test para cada una de las faltas de anclaje simples. Si llamamos N al número de nudos del circuito, el número de faltas que hay que considerar es $2N$ y el tiempo de generación de patrones de test será una función de $2N$ (si se consideraran las faltas múltiples, habría un total de $3^N - 1$ posibilidades). Además aún se puede reducir el tiempo de generación de test tomando sólo una falta de entre todas las faltas equivalentes [SCH72]. En la figura 1.10 se ilustra cómo el número de faltas de anclaje a considerar en una puerta AND de tres entradas se puede reducir teniendo en cuenta la equivalencia de faltas. Así, de un total de 8 faltas, se pasa a tener que considerar 5.

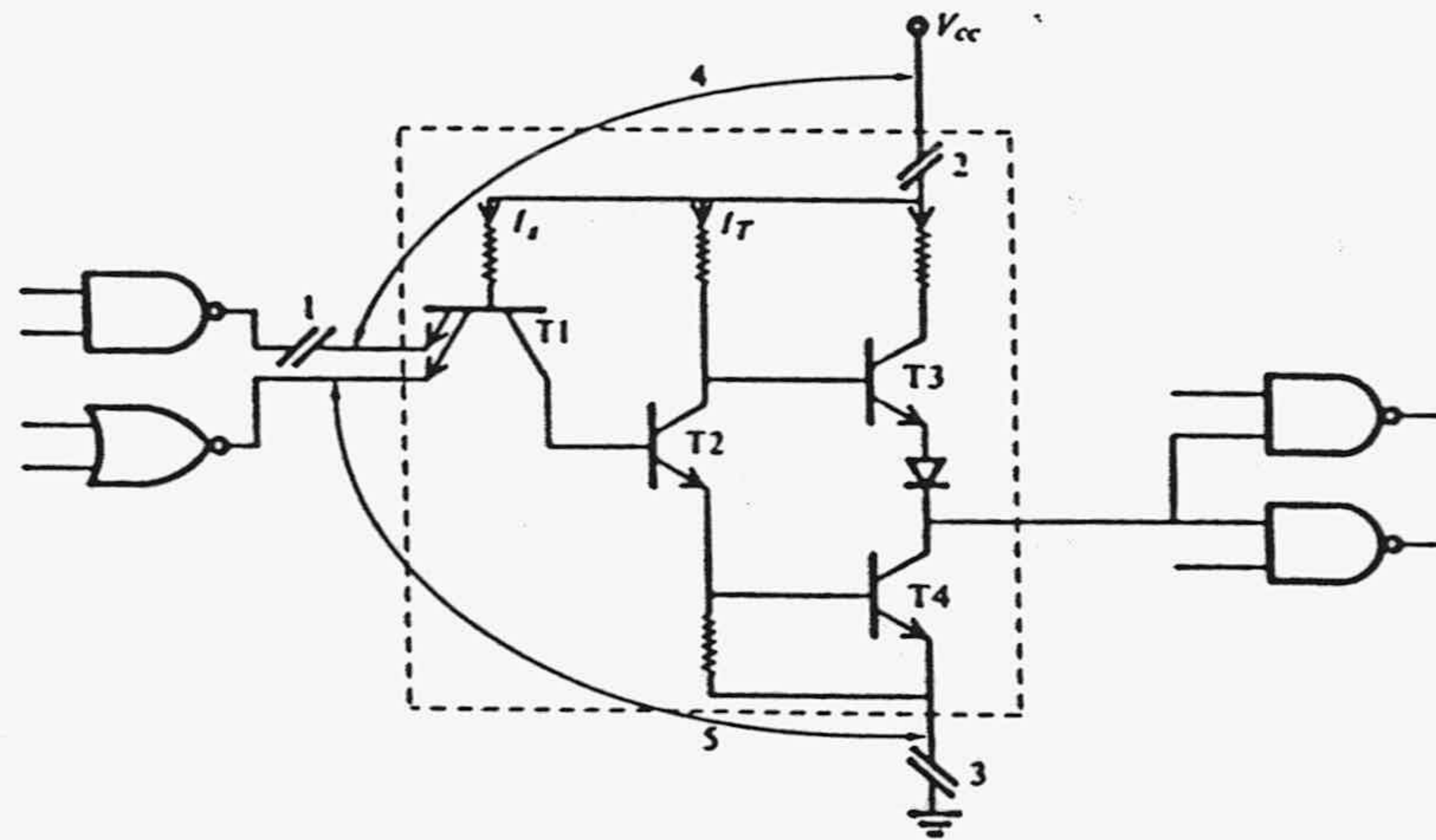


Figura 1.9. Algunos defectos de una puerta NAND TTL.

ABC	D	A(s-1)	B(s-1)	C(s-1)	D(s-1)	A,B,C,D(s-1)
000	0	0	0	0	1	0
001	0	0	0	0	1	0
010	0	0	0	0	1	0
011	0	1	0	0	1	0
100	0	0	0	0	1	0
101	0	0	1	0	1	0
110	0	0	0	1	1	0
111	1	1	1	1	1	0

Faltas Posibles: 8 (A,B,C,D s-0,s-1)
 Faltas No equivalentes: 5 (A,B,C,D s-1; A s-0)

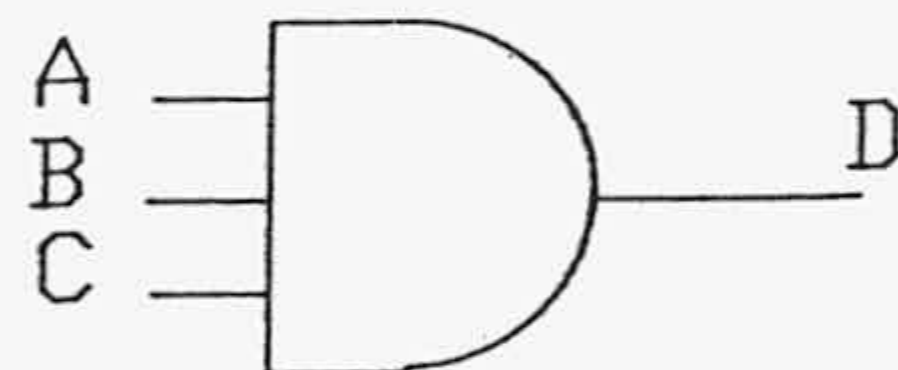


Figura 1.10. Faltas de anclaje en una puerta AND de tres entradas.

Otra forma de reducir el número de faltas a considerar parte del concepto de **dominancia de una falta por otra** [BRE76]: una falta f_1 se dice que domina a otra falta f_2 si todo patrón que detecte a f_2 detecta a f_1 , pero no a la inversa. Así, sólo es necesario buscar patrones para detectar f_2 , puesto que con ellos también se puede detectar f_1 .

En el nivel de descripción de puertas lógicas también se han utilizado las **faltas de cortocircuito o bridging** [MEI84, VOG78, KAR83]. Estas faltas corresponden a la conexión accidental de dos nudos del circuito. Como se muestra en la figura 1.11, según el tipo de lógica que se utilice, el cortocircuito se puede modelar como una puerta AND o una puerta OR adicional. Por otra parte, se pueden producir situaciones como la mostradas en la figura 1.12, que dan lugar a un comportamiento secuencial del circuito.

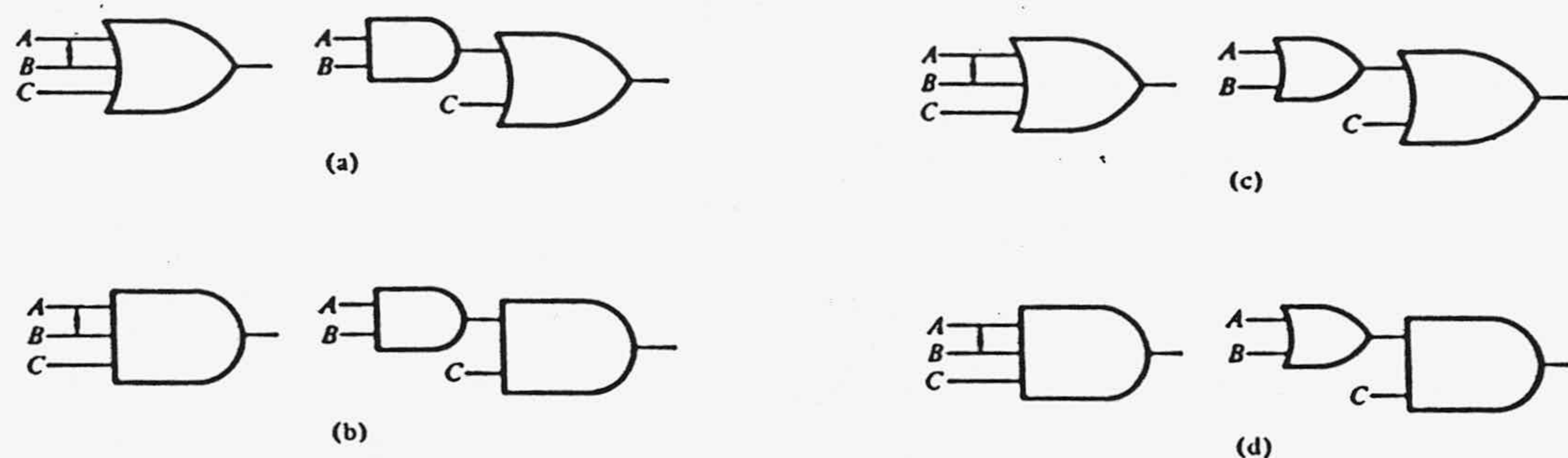


Figura 1.11. Faltas de cortocircuito (bridging) con lógica positiva (a) y (b), y negativa (c) y (d).

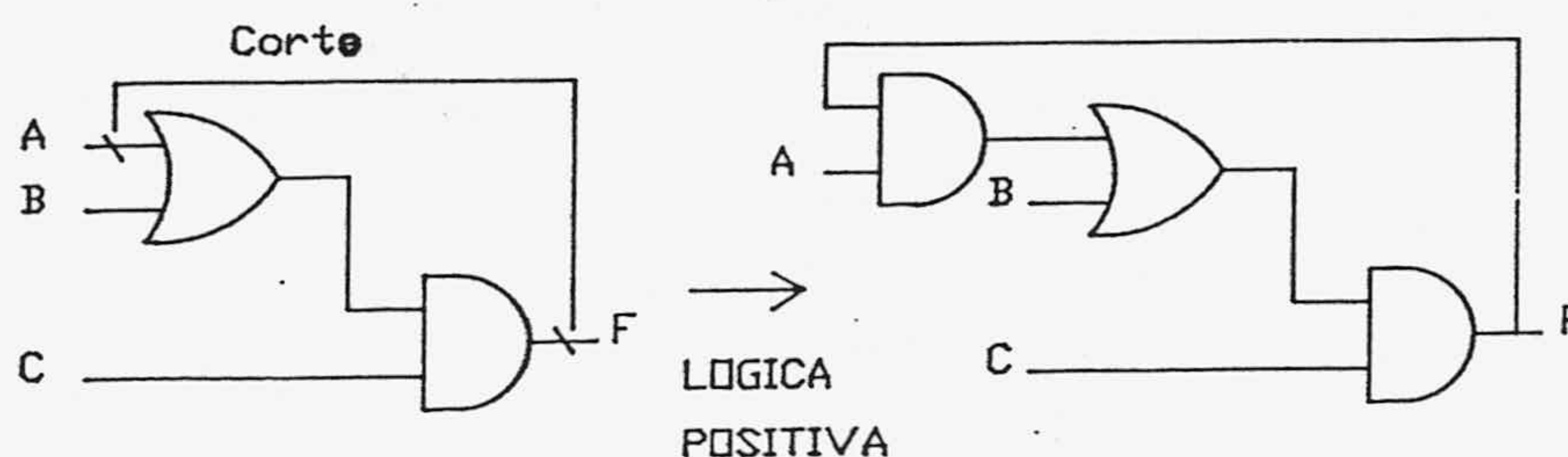


Figura 1.12. Falta de cortocircuito que hace secuencial un circuito combinacional.

Hasta ahora sólo se han considerado las faltas de anclaje y de cortocircuito. Tales faltas son incapaces de modelar ciertos defectos. Se ha observado [ALA87, BUR88, GAL80, RED84, SHE85] que, en los circuitos digitales MOS, existen defectos que no se pueden representar mediante faltas de anclaje y, lo que es peor, que los patrones de test generados para las faltas de anclaje no pueden detectar esos defectos: sólo entre un 10% y un 15% de esos defectos son detectados por tests generados para cubrir faltas de anclaje

[WAD78a, 78b]. Esos defectos se modelan considerando que los transistores MOS son conmutadores ideales y que pueden encontrarse permanentemente abiertos (transistor anclado abierto) o permanentemente cerrados (transistor anclado cerrado). La primera situación corresponde a las faltas denominadas de stuck-on y la segunda a las faltas de stuck-open.

Si el transistor T1 de la figura 1.5 presentase un defecto modelado como stuck-open, cuando se aplique $A=0, B=0$, el nudo Z no estará conectado ni a VDD ni a VSS, y se encontrará, por tanto, en un estado flotante o de alta impedancia, que hace que se pueda mantener el voltaje correspondiente a las anteriores entradas. Si Z se encontraba a 1, se mantendrá en tal estado y no se observará una salida errónea. Para detectar la falta habría que aplicar, antes de $AB=00$, la entrada $AB=10$, que asegurase que C esté a cero cuando se aplique el patrón de test $AB=00$.

Por otra parte, las faltas de transistor continuamente cerrado pueden ocasionar una tensión en las salidas que no corresponda a ninguno de los niveles lógicos, por lo que su detección caería fuera del marco de la verificación lógica del circuito.

Para tratar las faltas de stuck-open se han adoptado diversas estrategias. Por una parte, se ha propuesto asociar un modelo lógico a la red CMOS de transistores que se vaya a testear [JAI85, RED88]. En estos modelos el número de líneas que aparecen es mayor que el que existe en el circuito CMOS, como se puede comprobar en el modelo correspondiente a la puerta NOR CMOS mostrado en la figura 1.13, y cualquier falta de línea anclada o de transistor stuck-open se representa por una falta de anclaje en el modelo de puertas lógicas. Otra posibilidad es la de hacer el diseño del circuito de tal forma que las faltas en los transistores se puedan comprobar como si fueran faltas de anclaje de líneas [RED86, LIU87]. Por último, está la posibilidad de desarrollar simuladores y algoritmos para la generación de test que detecten este tipo de defectos [LO87, SCH84, JAI85b].

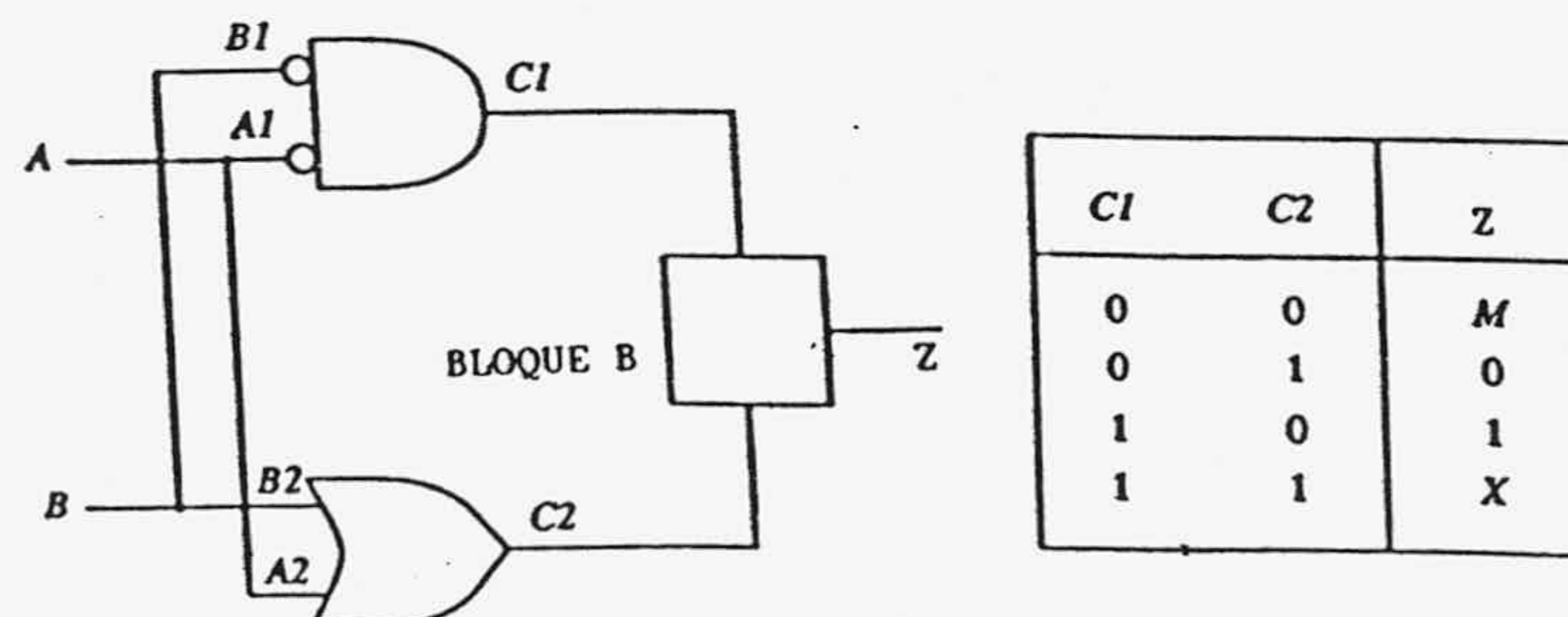


Figura 1.13. Modelo lógico de una puerta NOR CMOS para la descripción de faltas de anclaje en sus transistores (stuck-open y stuck-on), incluyendo la tabla de verdad del bloque B.

En determinados circuitos, como por ejemplo PLAs y memorias, además de los tipos de faltas que se han presentado, se utilizan otros modelos de defectos. Así, en PLAs es muy corriente utilizar las faltas de contacto o crosspoint [BOS82, ABR86b]. Estas modelan algunos de los defectos de una PLA como una pérdida de contacto (crosspoint⁻) o un contacto extra (crosspoint⁺) en alguno de sus planos. En la figura 1.14 se muestra un ejemplo de este tipo de faltas.

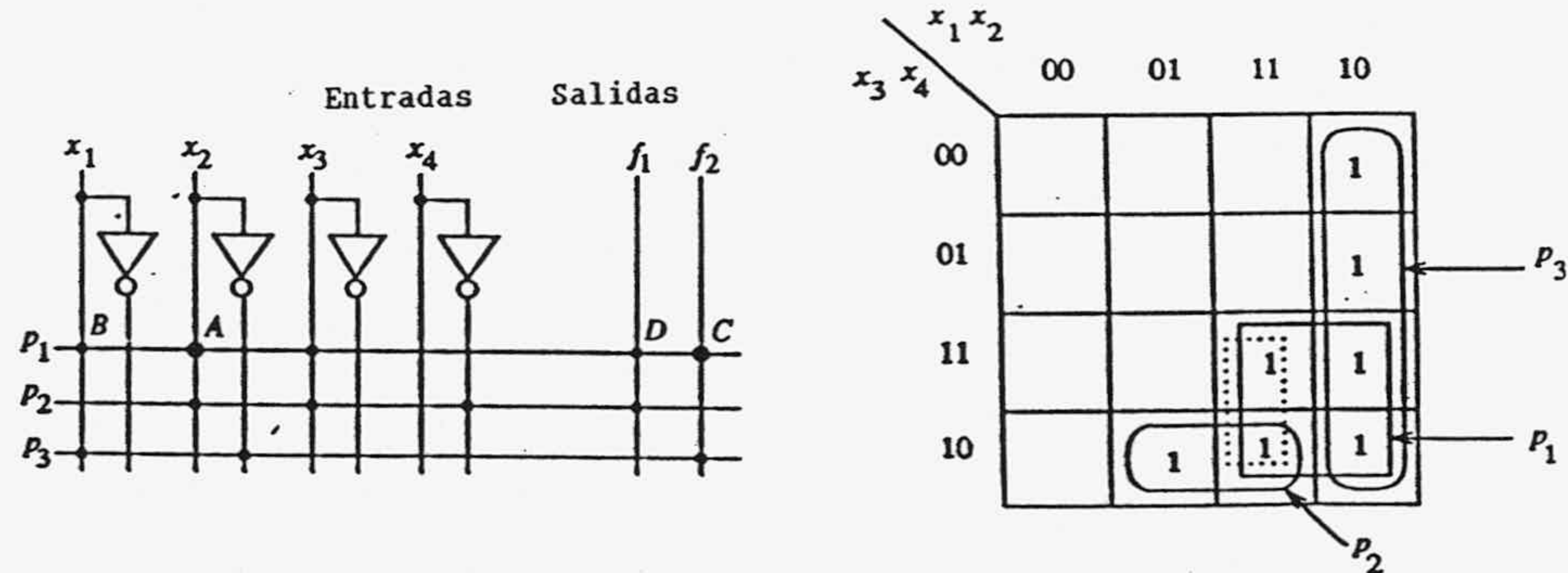


Figura 1.14. Faltas de contacto en una PLA. En el mapa de Karnaugh se observa (línea punteada) el efecto de la falta de contacto extra A en la línea p_1 .

Cuando se consideran las memorias, junto con las faltas de anclaje a 0 o a 1 de una celda, también se utilizan [ABA83, ABR86b] las faltas de acoplamiento entre celdas, que modelan los defectos considerando que los cambios en una celda pueden originar cambios en alguna de las adyacentes, y las faltas de sensibilidad a patrones [ABR86b], que modelan los defectos de las celdas suponiendo que el estado de dicha celda se modifica a causa de ciertos patrones de ceros, unos o transiciones en otras celdas.

Para concluir esta subsección, se va a considerar la posibilidad de que los defectos que presenta un circuito se modelen mediante faltas múltiples. En este caso, pudiera ocurrir que el error que ocasionaría un patrón de test para una falta fuera enmascarado por la existencia de otra falta. No obstante, se han desarrollado estudios [ABR80, BRE76, HUG86, JAC87] que muestran que un elevado porcentaje (más de un 99.6%) de las faltas múltiples de anclaje se detectan mediante test generados para las faltas de anclaje simples, y en cuanto a las faltas en circuitos CMOS se puede afirmar [JHA86] que el 99.6% de las faltas múltiples en una puerta NAND CMOS, con dominancia-n^(*), se pueden detectar por medio de los patrones generados para faltas simples y ese porcentaje aumenta con la complejidad de la puerta.

(*) La dominancia-n significa que la impedancia de la red de transistores nMOS es menor que la de la red de pMOS cuando están conduciendo.

1.2.4. Evaluación del test.

Para evaluar el test generado se suele utilizar el **cubrimiento de faltas**, es decir, la relación entre el número de faltas detectadas y el total de faltas que modelan los defectos físicos del circuito. Normalmente se supone que las faltas de anclaje simples son suficientes para modelar todos los defectos del circuito, aunque dadas las dimensiones tan reducidas de los circuitos es muy probable que un único defecto pueda ocasionar múltiples faltas. Sin embargo, considerar todas las posibles faltas múltiples es imposible dado su elevado número y, por otra parte, ya se vió en el apartado anterior que la mayoría de las faltas múltiples son detectadas por los mismos tests que se generan considerando únicamente faltas simples [ABR80, BRE76, HUG86, JAC87, JHA86]. Según esto, el cubrimiento de faltas puede ser útil a la hora de comparar distintos procedimientos de test, pero es discutible que proporcione una medida absoluta de la fracción de defectos físicos que se pueden poner de manifiesto con el procedimiento de test [AGR88, SHE85]. Para ello es necesario que exista una correspondencia biunívoca entre faltas y defectos, y eso no es totalmente cierto [ORT89, WIL81, SET84, SHE85]. No obstante, el **cubrimiento de faltas simples** sigue siendo una **figura de mérito** para un test. La razón para ello se encuentra en la relación existente entre el cubrimiento de faltas y una medida experimental de la efectividad de un test, que recibe el nombre de **nivel de defectos** [WIL81] o **relación de rechazo** [AGR88], definida como el cociente entre el número de circuitos defectuosos que han pasado el test frente al total de circuitos comprobados con dicho test. En [WAL78b] se propone la siguiente expresión para la relación de rechazo en función del cubrimiento de faltas:

$$r = (1 - CF)(1 - Y) \quad (1.2)$$

donde r es la relación de rechazo; CF es el cubrimiento de faltas; e $Y = \text{Circuitos Correctos} / \text{Circuitos Fabricados}$.

A partir de (1.2) se obtiene que si $CF=1$, $r=0$, y si $CF=0$ (no se aplica test alguno), $r=(1-Y)$, lo que significa aceptar como buenos todos los circuitos fabricados. Además (1.2) muestra que para un valor constante de Y , si se incrementa el valor de CF disminuye la relación de rechazo y, por consiguiente justifica el hecho de que los mejores test son los que proporcionan un mayor cubrimiento de faltas.

En [SET84] se presenta un modelo de la distribución de faltas en un

circuito integrado a partir de la distribución de defectos que se supone que afectan a dicho circuito. A partir de ese modelo, se determina una expresión para la relación de rechazo:

$$r=1-\left[\frac{1+B(1-e^{-c})}{1+B(1-e^{-cCF})}\right]^{-a} \quad (1.3)$$

en la que B , a y c son constantes que dependen del circuito y de su tecnología de fabricación y se determinan experimentalmente. El significado de estos coeficientes se puede obtener a partir de la comparación del rendimiento dado en [SET84]:

$$Y=[1+B(1-e^{-c})]^{-a} \quad (1.4)$$

con la ecuación de rendimiento de Stapper [STA75]

$$dY=[1-AD(\sigma/\mu)^2]^{-\mu/\sigma} \quad (1.5)$$

donde A es el área del circuito integrado, D es la media de la densidad de defectos y σ/μ es el coeficiente de variación de la densidad de defectos. Por consiguiente $\sigma/\mu=(1/a)^{1/2}$ y $AD=B(1-e^{-c})a$. A partir del modelo descrito en [SET84] se pueden construir curvas como la de la figura 1.15, que muestran que la expresión (1.2) constituye una buena aproximación para la determinación de la relación de rechazo.

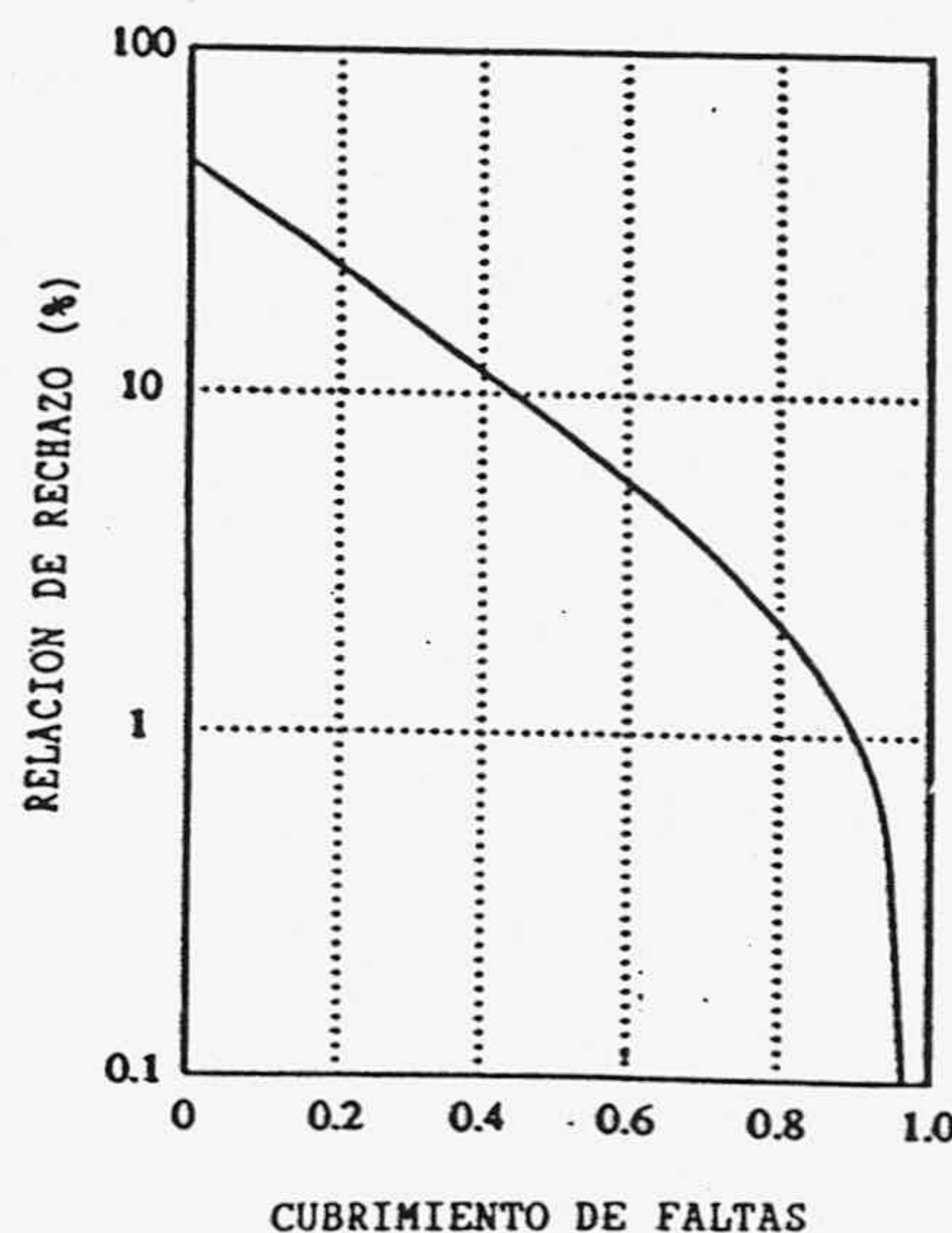


Figura 1.15. Dependencia de la relación de rechazo con el cubrimiento de faltas de anclaje simples.

Los resultados experimentales muestran que si se reducen las dimensiones del circuito manteniendo constantes el resto de parámetros, la curva de la figura 1.15 se desplaza hacia la izquierda [AGR88] y por consiguiente, cuanto mayor es la complejidad de los circuitos que se incluyen en un área de silicio dada, se precisa un cubrimiento de faltas menor para obtener una relación de rechazo dada. Por otra parte, si utilizamos (1.3), teniendo en cuenta el significado de sus parámetros dado en (1.5), se observa que, a medida que aumenta la superficie, A , el cubrimiento de faltas que se necesita para obtener una relación de rechazo dada tiende a un valor constante.

Basándose en la expresión (1.3) de la relación de rechazo y una vez determinados experimentalmente las constantes a , B , c para el circuito en cuestión, se pueden establecer las condiciones para aceptar un test en base a su cubrimiento de faltas según alcance o no un valor fijado para la relación de rechazo.

El cubrimiento de faltas se obtiene mediante la **simulación**. Se trata de simular el funcionamiento del circuito sin defectos para cada uno de los patrones de test, y su funcionamiento cuando se vea afectado por cada una de las faltas de anclaje simples (no equivalentes). Si para una falta, la salida del circuito para alguno de los patrones de test difiere de la que correspondería al circuito correcto, la falta (y todas sus equivalentes) se contabiliza como detectada. Según el algoritmo de simulación de faltas que se utilice, se habla de **simulación paralela** [SON85, THO75a, 75b], **simulación deductiva** [ARM72, MEN78] o **simulación concurrente** [ULR69, 73, 74]. En la simulación paralela se asocia un bit al circuito sin faltas y un bit a cada una de las faltas de anclaje simple, realizándose la simulación para todas las faltas, pero considerando en paralelo tantas como bits tenga una palabra del ordenador, menos una; en la simulación deductiva cada nudo tiene asociada una lista de todas aquellas faltas que, para cada valor lógico, ocasionan cambios en dicho valor, y el simulador deduce, para cada combinación de entradas, las listas de los nudos de salida a partir de las de los nudos de entrada; en la simulación concurrente, se realiza una simulación completa del circuito sin faltas y una simulación de la parte de circuito afectada por cada falta. A partir de la simulación concurrente se han desarrollado también simuladores como el FMOSSIM [SCH84], para circuitos descritos a nivel de conmutador, y el MOTIS [LO87], para circuitos descritos a nivel de puertas y de transistores (nivel de descripción mixto). La complejidad de los algoritmos de simulación es proporcional a G^2 para la simulación concurrente y deductiva, y proporcional a G^3 para la simulación paralela [CHA74, GOE80], siendo G el

número de puertas lógicas del circuito. Así pues, aún considerando únicamente las faltas simples de anclaje y no equivalentes, a medida que aumenta la complejidad del circuito, el tiempo de simulación puede ser extraordinariamente elevado. Es por ello que han aparecido métodos de muestreo de faltas [AGR81] y de análisis aproximado del cubrimiento de faltas, entre estos se encuentran el STAFAN (Statistical Fault Analysis) [JAI84, JAI85], el PPSFP (Parallel Pattern Single Fault Propagation) [WAI85] y el CPT (Critical Path Tracing) [ABR84].

A continuación, se considera uno de ellos a modo de ejemplo del tipo de información que proporcionan estos métodos. Concretamente se describe el que puede considerarse más popular de todos, el STAFAN. Este método utiliza la descripción del circuito a nivel de puertas y faltas simples de anclaje a 0 y a 1 en las líneas para la evaluación del cubrimiento de faltas. Mediante una única simulación del circuito sin faltas se determinan para todas las líneas (internas y externas) del circuito:

a) La controlabilidad de 1 (0) en la línea del circuito j , $C1(j)$ ($C0(j)$), definida como la probabilidad de que la línea j esté a 1 (0) cuando se aplica una combinación de entradas seleccionada aleatoriamente.

b) La observabilidad de un 1 (0) en la línea del circuito j , $B1(j)$ ($B0(j)$), que es la probabilidad de observar el valor a 1 (0) de la línea j del circuito en una de las salidas primarias. En el caso de una salida primaria $B0$ y $B1$ son iguales a 1.

A partir de los valores de $C0$, $C1$, $B0$ y $B1$ para cada línea del circuito, se pueden calcular las detectabilidades o probabilidades de detección de cada una de las faltas de anclaje simples cuando se aplica una entrada. Concretamente, la probabilidad de detectar una falta de anclaje a 1 en la línea j vendrá dada por

$$D1(j) = B0(j)C0(j) \quad (1.6)$$

y la de una falta de anclaje a 0 por

$$D0(j) = B1(j)C1(j) \quad (1.7)$$

Utilizando las detectabilidades obtenidas cuando se realiza la simulación del circuito con un conjunto de vectores prefijados se puede determinar el cubrimiento de faltas del circuito. Como sólo es necesaria una simulación sin faltas, la complejidad en el tiempo del procedimiento es proporcional al número de puertas G [JAI85].

1.2.5. Herramientas para la generación automática de patrones.

Para automatizar la generación de los patrones de test de los circuitos es necesario disponer de algoritmos que se puedan programar eficientemente. Sin embargo, se ha demostrado que el proceso de generación del conjunto óptimo de patrones es NP-completo [IBA75, FUJ82, FUJ90]. No obstante, hay que tener en cuenta que la dependencia exponencial en el tiempo de generación del test que se establece en [IBA75] se refiere al peor caso, es decir, al circuito en el que sea más complicada la búsqueda de un patrón de test. En la práctica se pueden obtener algoritmos correspondientes a heurísticas con las que se consigue, por término medio, un crecimiento menor del tiempo de generación de test con la complejidad del circuito. Este hecho se justifica a continuación.

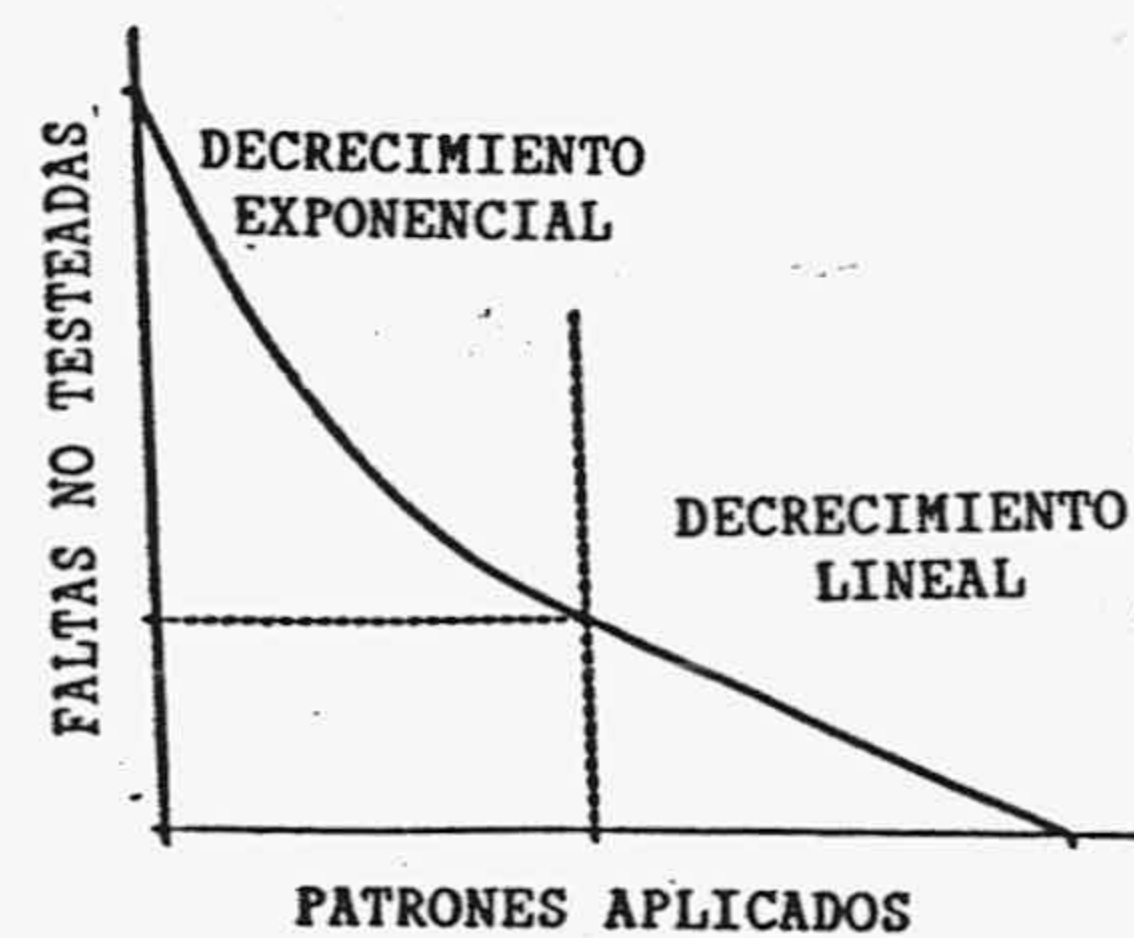


Figura 1.16. Dependencia entre el número de patrones aplicados y las faltas de anclaje no testeadas.

Experimentalmente se ha establecido que el número de faltas no detectadas a medida que se van aplicando patrones de test decrece exponencialmente hasta un número determinado de patrones aplicados, y linealmente a partir de ese momento [BAR81, GOE80]. En la figura 1.16 se muestra una curva típica del número de faltas no detectadas frente al número de test aplicados donde se refleja esa situación. Basándose en este hecho, y a partir de [GOE80] se puede establecer que el número de patrones de test en un circuito combinatorial crece linealmente con el número de puertas (G) y que el coste de generación del conjunto mínimo de patrones de test crece con el cuadrado del número de puertas (G^2). Así pues, es posible encontrar algoritmos que posibiliten la generación automática del test de un circuito para valores moderados de G .

La mayoría de los algoritmos que se utilizan hasta el momento se basan en el principio de la sensibilización de caminos [ARM66] en el circuito, es decir, tratan de encontrar una combinación de entradas que establezca un camino desde el nudo afectado por el defecto hasta alguna de las salidas. Entre los programas más utilizados para circuitos combinatoriales con faltas

a nivel de puertas lógicas se encuentran el algoritmo D [ROT67], el PODEM [GOE81], el FAN [FUJ83] y el CONT [TAK87]. También existen algoritmos aplicables para circuitos combinatoriales descritos a nivel de transistor como conmutador [BOS82a, RED84, 85], para circuitos secuenciales [MAL85, MUT76, THO71, SNE77] y para PLAs [BOS82b]. No obstante, para los circuitos secuenciales, los generadores de tests no son tan efectivos y se utilizan otras técnicas que se referirán en las secciones 1.3 y 1.6.

1.2.6. Estimación de la longitud del test pseudoaleatorio.

La generación aleatoria de patrones [AGR75, MCL88, WAG87] permite una rápida selección del conjunto de combinaciones de entradas que constituyen el test. Sin embargo, el problema aparece cuando se trata de determinar el cubrimiento de faltas que se consigue con los patrones seleccionados [MAL84] o el número de patrones, L , que hay que elegir para asegurar un cubrimiento de faltas suficiente [SAV84b, WAG87]. Una forma de evitar esta dificultad se basa en el uso de la simulación de faltas que se vió en el anterior apartado. En este caso, dado un patrón seleccionado aleatoriamente, se determina mediante simulación qué faltas detecta. Esas faltas se eliminan del conjunto de faltas a simular y se prosigue para el siguiente patrón hasta que no queda ninguna falta por detectar o se alcanza el cubrimiento deseado [MAL84].

Por otra parte, se han obtenido expresiones que permiten evaluar la longitud de un test aleatorio (número de patrones seleccionados aleatoriamente) a aplicar en función de las **detectabilidades de las faltas de anclaje simples** [CHI87, SAV84b, WAG87]. La detectabilidad de una falta se define como la relación entre el número de combinaciones de entradas que pueden ponerla de manifiesto y el número de combinaciones posibles. Por ejemplo, en [WAG87] se obtienen expresiones para el **cubrimiento de faltas esperado**, $E(C_L)$, y para la **longitud del test aleatorio**, L , respectivamente:

$$E(C_L) = 1 - \sum_{k=1}^{N-L} (h_k/M) \{ [(N-L)!k!] / [N!(N-L-k)!] \} \quad (1.8)$$

$$L = N(1 - (1 - C_L)^{1/k_m}) \quad (\text{para } N-L \gg km) \quad (1.9)$$

donde C_L es el cubrimiento de faltas, N es el número de combinaciones de entradas posibles, M es el número de faltas posibles, h_k es el número de faltas con detectabilidad k , y km es la detectabilidad más baja para alguna de las faltas del circuito. La cuestión a resolver es, por consiguiente, la

obtención eficiente de las detectabilidades para las faltas del circuito, es decir los valores de k , ya que a partir de ellas se pueden determinar los valores de h_k y evaluar (1.8) y (1.9).

Teniendo en cuenta los conceptos de **observabilidad** y **controlabilidad**, presentados en la sección 1.2.2, debe existir alguna relación entre estos y la detectabilidad. Es lógico pensar, atendiendo a las definiciones, que las faltas de anclaje en un nudo con mucha controlabilidad y observabilidad presenten altas detectabilidades.

Puesto que en un circuito con testeabilidades elevadas en sus nudos, la longitud del test aleatorio será pequeña, interesa aumentar estas. Por consiguiente, debido a la relación entre detectabilidad y testeabilidad, será beneficioso realizar diseños con altos valores de testeabilidad en todos sus nudos. En [RED72, BEN84] se da una serie de reglas para conseguir este propósito y en [BRG89a, YOU89] se presentan sendos procedimientos de síntesis automática de circuitos con elevada testeabilidad. Otros trabajos muestran diseños fácilmente testeables de distintos circuitos como multiplicadores [SHE84], PLAs [KHA84, FUJ81, RED87, RAJ87, SAL83], etc. No obstante, hay que tener en cuenta que no siempre se puede identificar detectabilidad y testeabilidad. Como se demuestra en [SAV83], una alta detectabilidad y controlabilidad en un nudo no siempre implican altas detectabilidades de las faltas de anclaje asociadas.

Las expresiones dadas en [SAV83] para la controlabilidad, la observabilidad y la detectabilidad de una falta de anclaje en la línea g del circuito que sintetiza las m funciones $F=(F_1, \dots, F_m)$ de n variables $Y=(y_0, \dots, y_{n-1})$ son las siguientes:

$$c(g/0)=S(g(Y)) \quad (1.10)$$

$$c(g/1)=S(g'(Y)) \quad (1.11)$$

$$o(g/0)=o(g/1)=S(U_{j=1}^m \delta F_j / \delta g) \quad (1.12)$$

$$t(g/0)=S(g(Y)U_{j=1}^m \delta F_j / \delta g) \quad (1.13)$$

$$t(g/1)=S(g'(Y)U_{j=1}^m \delta F_j / \delta g) \quad (1.14)$$

donde $c(g/i)$ es la fracción de combinaciones de entrada que hacen que la línea g esté a i' (controlabilidad de la falta de anclaje a i en g), $o(g/i)$ es la fracción de combinaciones de entrada que propagan el valor de la línea g a alguna de las m salidas del circuito (observabilidad de una falta de anclaje a i en g) y $t(g/i)$ es la fracción de entradas que ponen de manifiesto la falta de anclaje a i en g , es decir, la **detectabilidad de la falta**. En (1.10)-(1.14), $S(g(Y))$ es la fracción de entradas para las que la función g es uno

y recibe el nombre de síndrome, $\delta F_j / \delta g$ es $F_j(g=0) \circ F_j(g=1)$ y U es la suma booleana (OR).

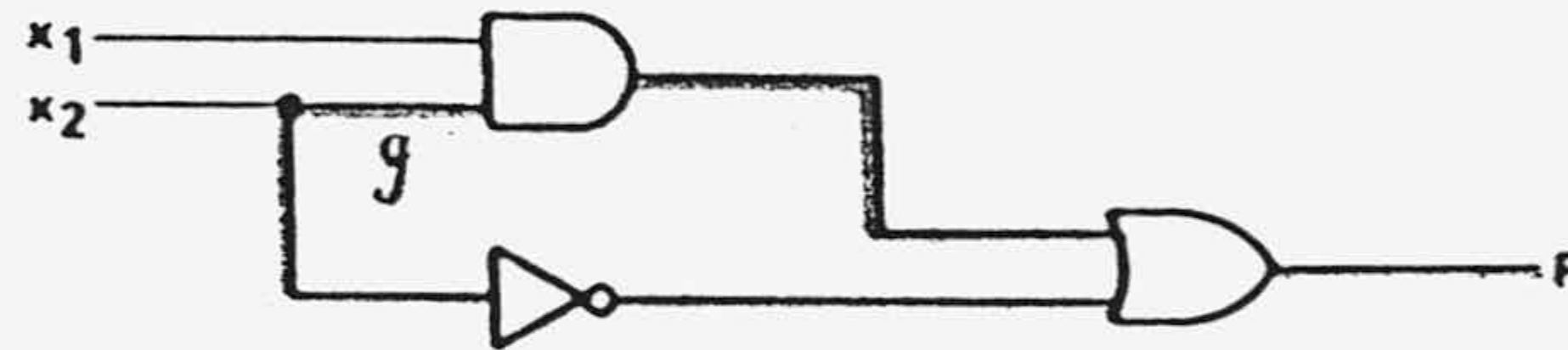


Figura 1.17. Ejemplo de circuito lógico.

Si se considera el circuito de la figura 1.17, se observa que $c(g/1) = S(y_1') = 1/2$, $o(g/1) = S(y_0, y_1) = 1/4$ y sin embargo, $t(g/1) = S(y_0 y_1 y_1') = 0$. También hay ejemplos en los que, con observabilidades y controlabilidades bajas, una falta tiene una elevada detectabilidad (figura 1.18). El problema para identificar controlabilidad y observabilidad con detectabilidad se debe a la existencia de nudos de fanout reconvergente o reconvergencias [SAV84a, MAA87]. Una reconvergencia es aquel nudo que actúa como entrada a varias puertas lógicas cuyas salidas influyen en otro nudo del circuito, a través de una o varias de puertas lógicas. En las figuras 1.17 y 1.18 se muestran ejemplos de reconvergencias ; con trazo grueso se han marcado los caminos de reconvergencia.

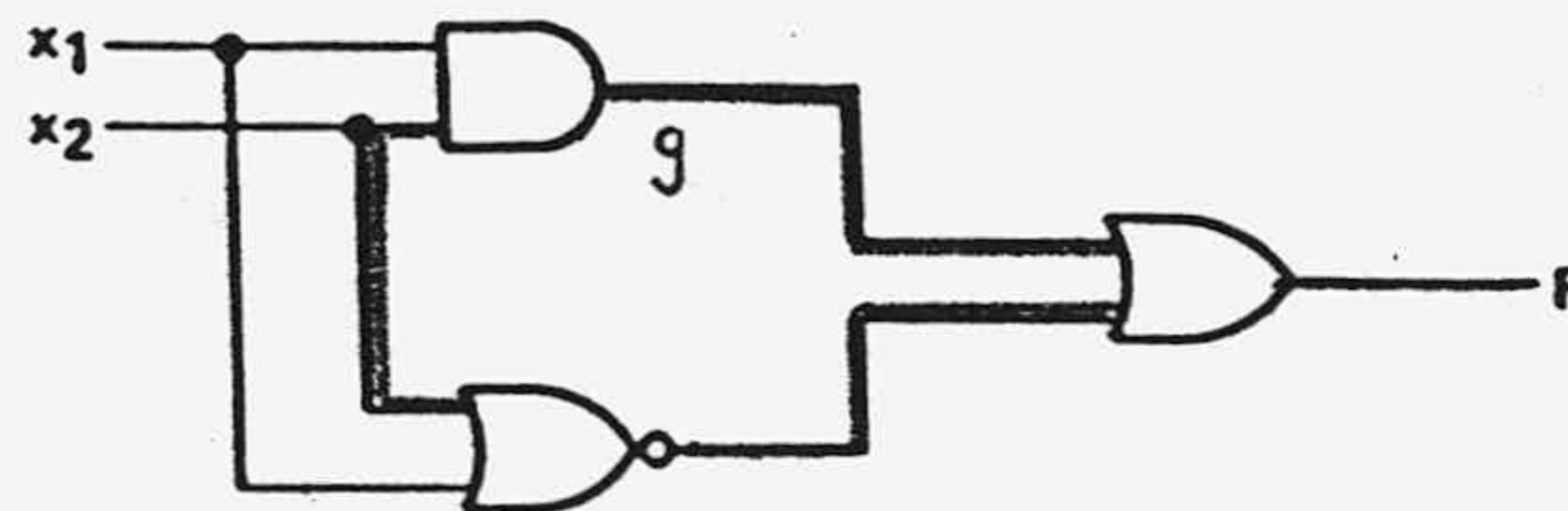


Figura 1.18. Circuito con elevada detectabilidad.

Otro procedimiento a considerar para obtener la testeabilidad de un circuito frente a patrones de test aleatorios es el algoritmo de corte [SAV84a], que asocia a cada nudo del circuito la probabilidad de que se encuentre en cada valor lógico; su nombre se debe a que las reconvergencias se cortan, apareciendo tantos nudos nuevos como líneas de reconvergencia, y se les asocia no una probabilidad sino un intervalo de probabilidades. Además, existen programas para la evaluación de las detectabilidades en los nudos (COP [BRG84] y PREDICT [SET85]) y para calcular las probabilidades de obtener un valor dado en las salidas del circuito [PAR75a, 75b].

1.3. Las soluciones al problema de accesibilidad de las entradas.

Para poder aplicar los patrones de test a módulos con entradas no accesibles externamente se realizan ciertas modificaciones en los mismos [WIL83], en dos aspectos:

- añadiendo la circuitería necesaria para la generación automática de los patrones de test y para el análisis de las respuestas a los mismos (figura 1.19), y
- haciendo fácilmente accesibles los biestables de los circuitos secuenciales (figura 1.20).

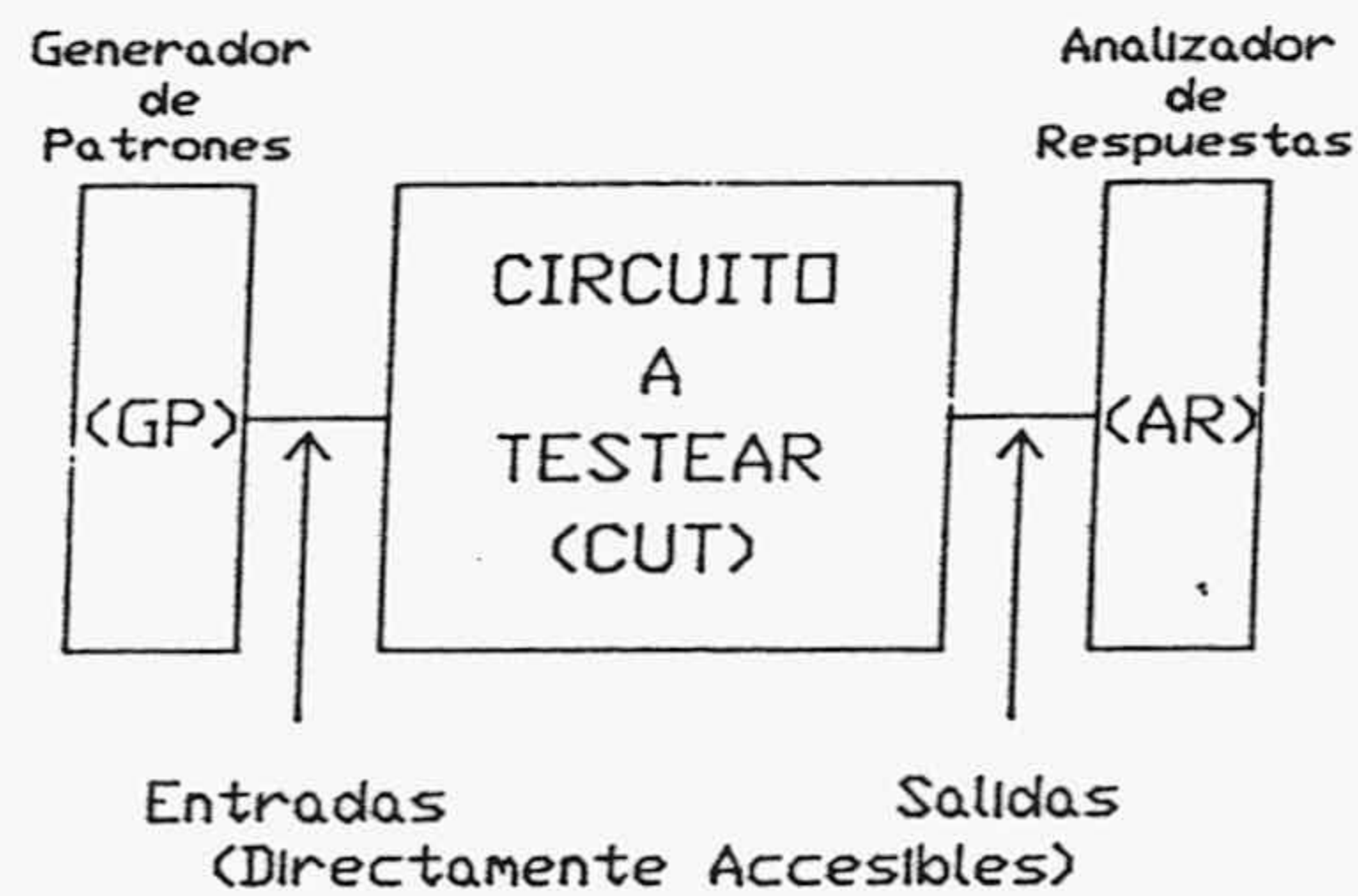


Figura 1.19. Circuito a testear (CUT) con los módulos de generación de patrones de test (GP) y de análisis de la respuestas a aquellos (AR).

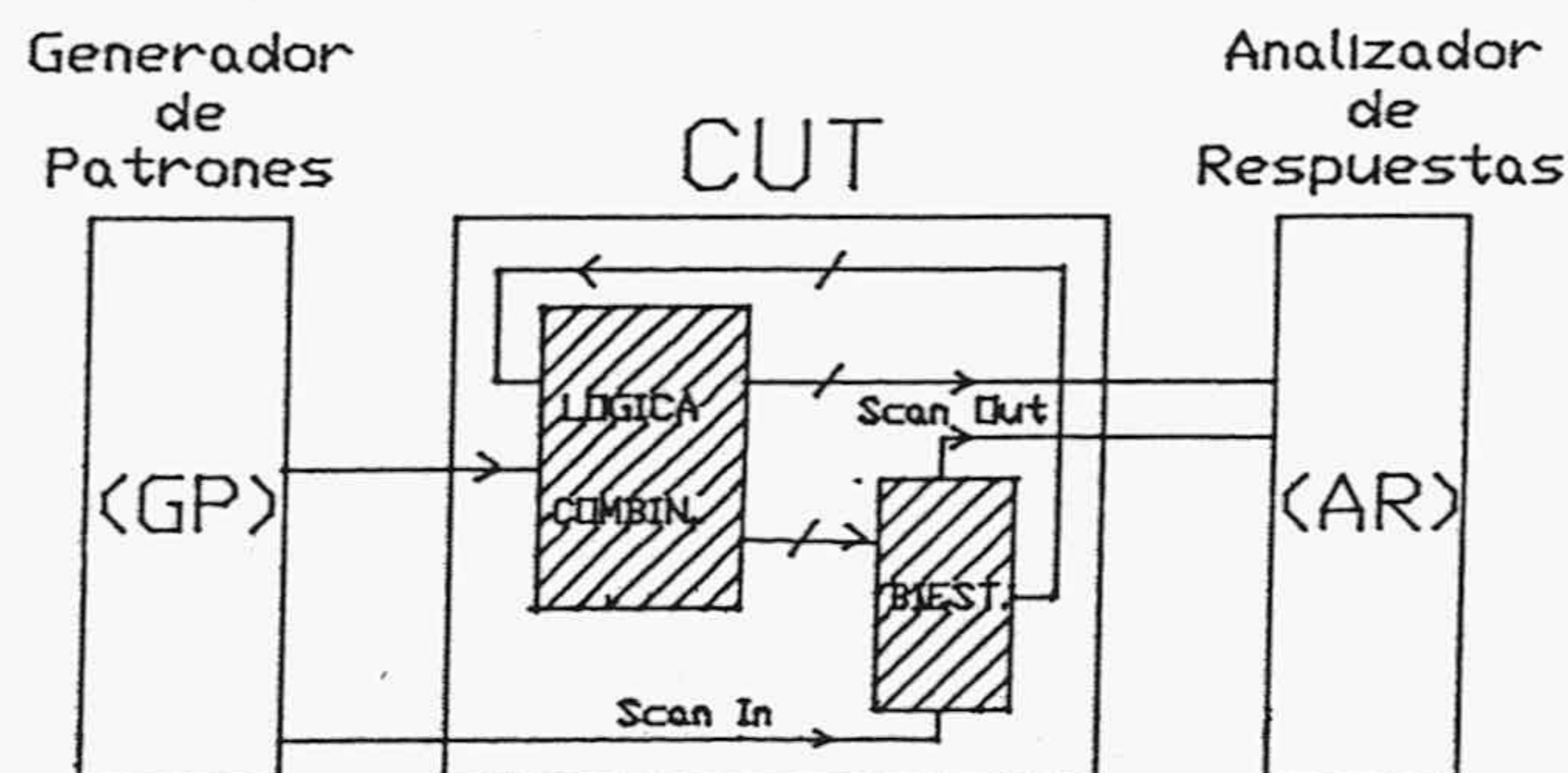


Figura 1.20. Modificaciones para acceder al estado de un CUT secuencial. Los biestables se organizan de manera que en el modo de test se puedan actualizar y leer como en un registro de desplazamiento.

Si el procedimiento de test es exhaustivo o pseudoexhaustivo, la aplicación automática de los patrones de test se puede conseguir mediante contadores binarios. En el caso de test exhaustivo habría un contador de tantos bits como entradas tuviera el circuito, y en el caso de test pseudoexhaustivo habría tantos contadores como subcircuitos, cada uno con el

número de bits correspondiente al de entradas del subcircuito.

Si el procedimiento de test utiliza patrones seleccionados aleatoriamente, se puede utilizar un registro de desplazamiento con realimentación lineal, LFSR (Linear Feedback Shift Register) [CHE88, TSU87], con tantos biestables como entradas tenga el circuito. Además, mediante un LFSR también se pueden generar todas las combinaciones posibles de entradas, necesarias en el test exhaustivo [WANG88]. En la figura 1.21 se muestran una serie de ejemplos de LFSRs junto con el polinomio definido en el cuerpo de Galois, $GF(2)$, en el que se basan. Si ese polinomio es un polinomio primitivo [LIN83], el LFSR puede generar todas las combinaciones de entrada excepto la $(0,0,\dots,0)$ si se encuentra inicialmente en un estado distinto del $(0,0,\dots,0)$ [LIN83, TAN84]. En la Tabla 1.1 aparecen algunos polinomios primitivos y la forma de obtener el correspondiente LFSR.

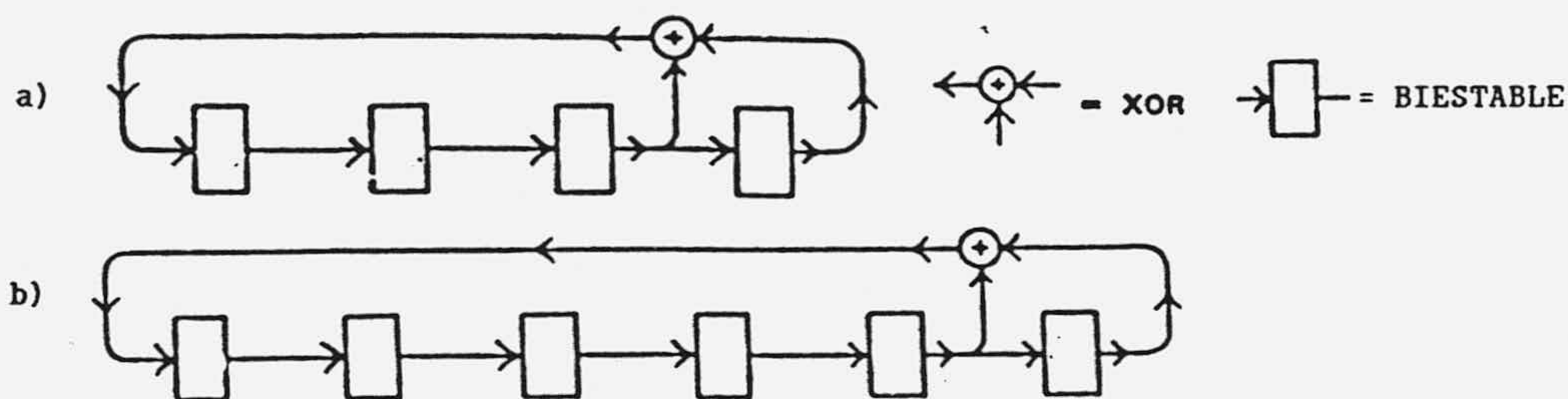


Figura 1.21. Ejemplos de LFSRs basados en: a) x^4+x+1 ,
b) x^6+x+1 .

En el caso de que los patrones de test se seleccionen según un procedimiento determinista o tengan que aplicarse según una secuencia fija, el circuito de aplicación habrá de diseñarse en cada caso concreto y, según su complejidad, se podrá integrar junto con el bloque a testear o no. Los procedimientos de test en los que se añaden al circuito los módulos necesarios para testearlo de manera automática, o autónoma, reciben el nombre de procedimientos de Autotest, con frecuencia se designan con las siglas BIST o BIT (Built-In Self Test) [BAR82, BEN75, CRO80, DAN85, HUA84, MCL81, 85a, 85b, YOU89]. En [GRA82, TRE85, TRE87] se pueden consultar algunos de estos diseños BIST para PLAs y en [THA82, KUB84, GEL87] se muestra su utilización en el test de microprocesadores.

En cuanto a la modificación del circuito para poder acceder a los biestables, existen procedimientos como las técnicas de Scan Path o de Estado Directamente Accesible (EDA) [WIL73, EIC77, FUN78, AND80] que modifican el

diseño de los circuitos secuenciales para que, al aplicar el test, se pueda acceder a los biestables como si de entradas primarias se tratase. Dentro de esas técnicas se pueden incluir las estructuras LSSD [DAS80, 82]. En la figura 1.22 se muestra un ejemplo de este tipo de diseños.

TABLA 1.1. Polinomios Primitivos y obtención de LFSRs.

GRADO	MINIMO POLINOMIO PRIMITIVO	REALIMENTACION (izqda. a drcha)
2	1, 2	X^2+X+1
3	1, 3	X^3+X+1
4	3, 4	X^4+X+1
5	3, 5	X^5+X^2+1
6	5, 6	X^6+X+1
7	4, 7	X^7+X^3+1
8	2, 3, 5, 8	$X^8+X^6+X^5+X^3+1$
9	5, 9	X^9+X^4+1
10	7, 10	$X^{10}+X^3+1$
11	9, 11	$X^{11}+X^2+1$
12	6, 8, 11, 12	$X^{12}+X^6+X^4+X+1$
13	9, 10, 12, 13	$X^{13}+X^4+X^3+X+1$
14	4, 8, 13, 14	$X^{14}+X^{10}+X^6+X+1$
15	14, 15	$X^{15}+X+1$
16	4, 13, 15, 16	$X^{16}+X^{12}+X^3+X+1$
17	14, 17	$X^{17}+X^3+1$
18	11, 18	$X^{18}+X^7+1$
19	14, 17, 18, 19	$X^{19}+X^5+X^2+X+1$
20	17, 20	$X^{20}+X^3+1$
21	19, 21	$X^{21}+X^2+1$
22	21, 22	$X^{22}+X+1$
23	18, 23	$X^{23}+X^5+1$
24	17, 22, 23, 24	$X^{24}+X^7+X^2+X+1$
25	22, 25	$X^{25}+X^3+1$
26	20, 24, 25, 26	$X^{26}+X^6+X^2+X+1$
27	22, 25, 26, 27	$X^{27}+X^5+X^2+X+1$
28	25, 28	$X^{28}+X^3+1$
29	27, 29	$X^{29}+X^2+1$
30	7, 28, 29, 30	$X^{30}+X^{23}+X^2+X+1$
31	28, 31	$X^{31}+X^3+1$
32	10, 30, 31, 32	$X^{32}+X^{22}+X^2+X+1$
33	20, 33	$X^{33}+X^{13}+1$
34	7, 32, 33, 34	$X^{34}+X^{27}+X^2+X+1$

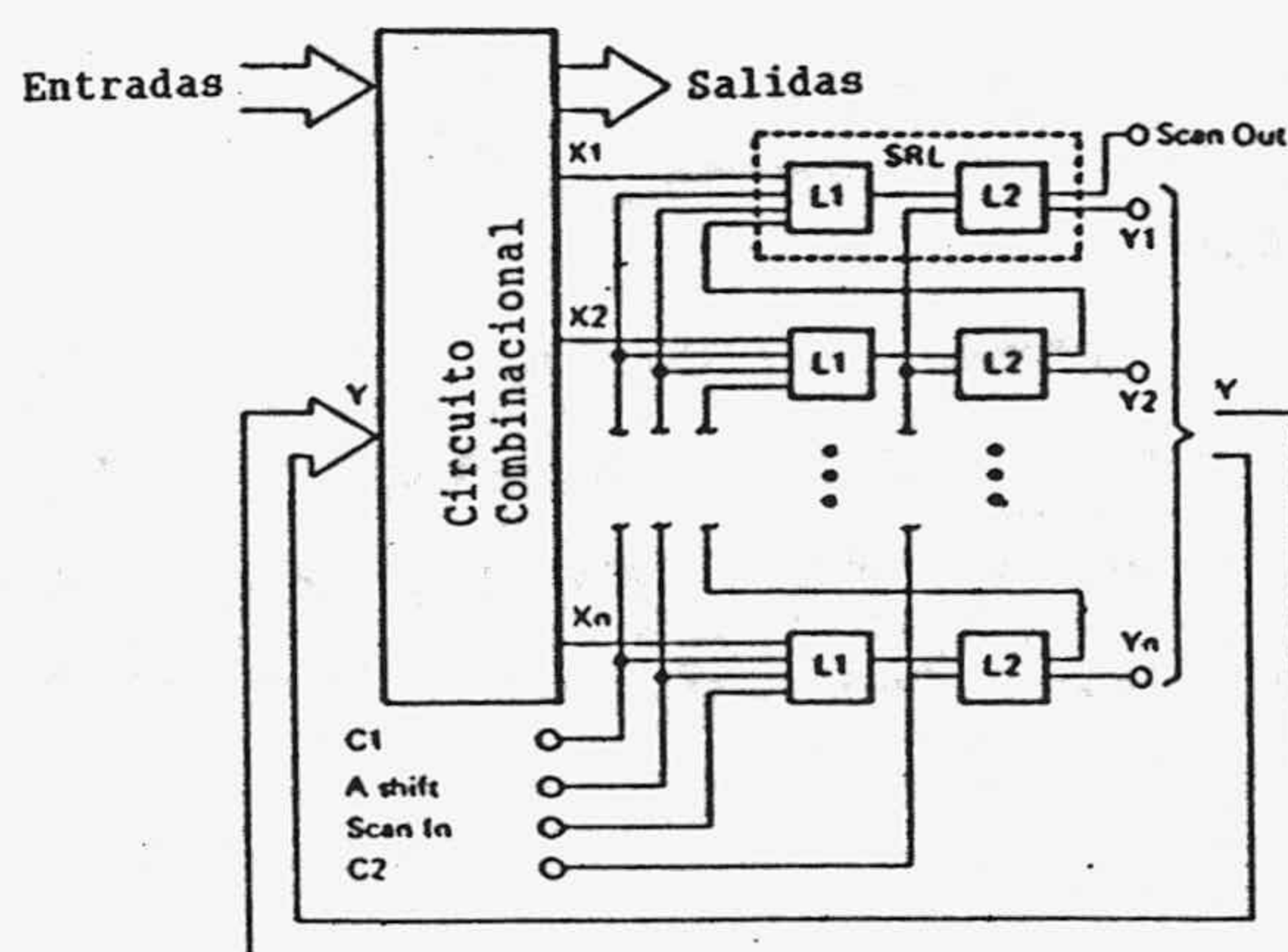


Figura 1.22. Circuito secuencial modificado según un diseño LSSD para acceder a su estado en el test.

1.4.El problema del Análisis de las Respuestas.

Para detectar la presencia de un defecto es necesario comparar las salidas, o alguna función de éstas, con las que cabría esperar para el circuito correcto. Si la comparación se hiciera utilizando las salidas obtenidas para cada entrada, se necesitaría una memoria con las salidas correctas o de un procedimiento software o hardware (una "copia" correcta del circuito a testear) para generar las salidas correctas a cada entrada. Si el número de patrones de test es elevado y se utiliza una memoria, ésta habrá de ser de una gran capacidad. Para solventar este problema se puede recurrir a la **compresión de las salidas del test** [ROB87]. Es decir, se define una función que genera una descripción de las respuestas a los patrones de test con menos bits de los que se necesitarían para la descripción exacta de las mismas: la **signatura o firma** [HAS84, IWA88]. Los procedimientos que utilizan esta solución reciben también el nombre de procedimientos de análisis de firmas o **signaturas**.

Sea un circuito, R^F , que sintetiza n funciones de m variables, f_1, f_2, \dots, f_n , notadas con F , $F=(f_1, f_2, \dots, f_n)$, y con entradas (y_1, \dots, y_m) , notadas con Y , $Y=(y_1, \dots, y_m)$. La secuencia de patrones de test, Y^1, Y^2, \dots, Y^L , se notará con Y^T , y la secuencia de salidas que se obtienen, $F(Y^1), F(Y^2), \dots, F(Y^L)$, con F^T .

Una signatura para el circuito R^F es un conjunto de r bits, s_1, \dots, s_r , ($s_i \in C_2$) que se obtienen a partir de las salidas del circuito para la secuencia de entradas, Y^1, Y^2, \dots, Y^L , que constituyen el test del circuito mediante un procedimiento de compresión, S_r , de manera que

$$S_r(F(Y^1), F(Y^2), \dots, F(Y^L)) = S_r(F^T) = (s_1, \dots, s_r) \quad (1.15)$$

para la secuencia de salidas del circuito sin defectos. La signatura constituye el resultado del test: comprobando su valor se determina si el circuito funciona correctamente o no.

Si el circuito presenta un defecto detectable por la secuencia de test para alguno de los patrones, Y^j , se observará una salida, $F^*(Y^j)$, diferente a la salida esperada, $F(Y^j)$:

$$F^*(Y^j) = F(Y^j) \oplus E^j \quad (1.16)$$

donde la desviación de $F^*(Y^j)$ respecto de $F(Y^j)$, E^j , es el patrón de error.

El patrón de error $E^j = (e_1^j, \dots, e_N^j)$ se construye a partir de las desviaciones, e_k^j ($k=1, \dots, N$), observadas en las N funciones que sintetiza el circuito, $f_k(Y^j)$ ($k=1, \dots, N$)

$$f_k(Y^j) \oplus e_k^j. \quad (1.17)$$

Según la función de compresión definida se pueden distinguir diversos procedimientos de test. Así tenemos los procedimientos de **análisis de síndrome** [SAV80, SAV81, SER86, TUN88], procedimientos **espectrales** [ERI86, HSI86, KAR81, MIL, RUI89, SUS83, DAR89], procedimientos basados en los **códigos residuos** [DAV80, SAV88, WIL88, PRA90], procedimientos de **compresión cuadrática** [KAR90], procedimientos de **cuenta de transiciones** [ROB87, HAY78, HUR86], etc.. Estos procedimientos presentan la posibilidad de **aliasing** o **enmascaramiento** [ROB87], es decir, el hecho de que circuitos defectuosos pueden presentar, una vez aplicado el test, la misma signatura que el circuito correcto. Por consiguiente, si se pretende adoptar una solución basada en un procedimiento de compresión habrá que:

- a) evaluar la posibilidad de enmascaramiento que, para un circuito concreto, presenta el procedimiento de compresión elegido [DAV80, DAV90, PRA90, SMI80, WIL88], o
- b) elegir el procedimiento para el que se minimice la posibilidad de enmascaramiento en un circuito dado [KAR90, ROB87].

Tanto en un caso como en otro es necesario disponer de información de la posibilidad de observar ciertos patrones de error en un circuito defectuoso cuando se aplica un patrón de test. Es lo que se llaman **modelos de error** [ABR86b, GUP88]. Existen trabajos que, a partir de diversos modelos de errores, permiten estimar la probabilidad de enmascaramiento para los diferentes métodos de compresión [DAV80, GUP88, ROB87, SMI80]. A continuación se presenta una expresión general para evaluar el enmascaramiento.

1.4.1. Evaluación del enmascaramiento.

En esta sección se establece una expresión que permite evaluar la probabilidad de enmascaramiento. El problema del enmascaramiento consiste en que aún en presencia de un defecto detectable por la secuencia de test, se obtiene la signatura del circuito correcto, es decir

$$S_r(F^T \oplus E^T) = S_r(F^T) = (s_1, \dots, s_r) \quad (\text{con algún } E^j \neq 0, j=1, \dots, L)$$



Dado un procedimiento de compresión, S_r , y una secuencia de patrones de test, Y^T , para una función $F(Y)$, se define la **función de enmascaramiento** para el procedimiento de compresión S_r , $A(F^T, E^T)$, como aquella que cumple

$$\begin{aligned} A(F^T, E^T) &= 1 \text{ si } S_r(F^T) = S_r(F^T \oplus E^T) \\ A(F^T, E^T) &= 0 \text{ si } S_r(F^T) \neq S_r(F^T \oplus E^T) \end{aligned} \quad (1.19)$$

Por consiguiente, la probabilidad de que la secuencia de test aplicada, Y^T , no detecte un defecto en el circuito debido a la compresión que se lleva a cabo en las salidas (probabilidad de enmascaramiento) viene dada por la expresión:

$$P_A = \sum_{E^T \neq 0} A(F^T, E^T) p(E^T / F^T) \quad (1.20)$$

donde F^T es la secuencia de salidas que se obtienen en el circuito correcto para la secuencia de patrones de test, y $p(E^T / F^T)$ la probabilidad de que se produzca la secuencia de errores E^T cuando el circuito sin defectos ha de generar la secuencia de salidas F^T . Con respecto a la expresión (1.20) se pueden hacer las dos consideraciones siguientes:

a) Esa expresión permite evaluar la probabilidad de enmascaramiento del procedimiento de compresión aplicado a un circuito concreto y para cualquier secuencia de patrones de test que genere la secuencia de salida F^T . Si se desea evaluar la probabilidad de enmascaramiento de un procedimiento de compresión para distintos circuitos y para todas las secuencias de salidas F^T posibles, hay que extender la suma de (1.20) a los posibles valores de dichas secuencias:

$$P_A = \sum_{F^T} p(F^T) \{ \sum_{E^T \neq 0} A(F^T, E^T) p(E^T / F^T) \} = \sum_{Y, E^T \neq 0} A(F^T, E^T) p(F^T, E^T) \quad (1.21)$$

b) Para evaluar (1.20) y (1.21), es necesario determinar la función de enmascaramiento, A , del procedimiento de compresión y conocer las probabilidades $p(F^T, E^T)$ para todas y cada una de las secuencias de patrones de error y de salidas observables en el circuito sin defectos.

El problema de la determinación del **procedimiento de compresión óptimo** significa encontrar el procedimiento para el que se **función de enmascaramiento minimiza P_A** , una vez determinadas las probabilidades $p(F^T, E^T)$ (en el Apéndice I se considera el enmascaramiento en LFSRs de varias entradas).

1.5. Las soluciones al problema de observabilidad de las salidas.

En los circuitos VLSI, es frecuente que un módulo se encuentre dentro de un circuito integrado, de forma que sus salidas no sean accesibles exteriormente. Es preciso, por consiguiente, añadir la circuitería que se encargue de analizar las respuestas a los patrones de test, más concretamente se analiza la signatura obtenida por el procedimiento de compresión utilizado. La cuestión que se plantea es la de determinar, para cada circuito, qué procedimiento de compresión necesita una circuitería de análisis más simple, verificando unas restricciones dadas en la posibilidad de enmascaramiento.

Un esquema que se ha venido utilizando con bastante frecuencia es el procedimiento de análisis de códigos residuos, realizado mediante un LFSR. Este registro se puede modificar para que también pueda utilizarse como generador de los patrones de test. Un circuito de tales características recibe el nombre de BILBO (Built-In Logic Block Observer) o BIDCO [FAS80, KON79, WIL83], mostrándose en las figuras 1.23 y 1.24 la estructura de este circuito y la forma de utilizarlo para el test.

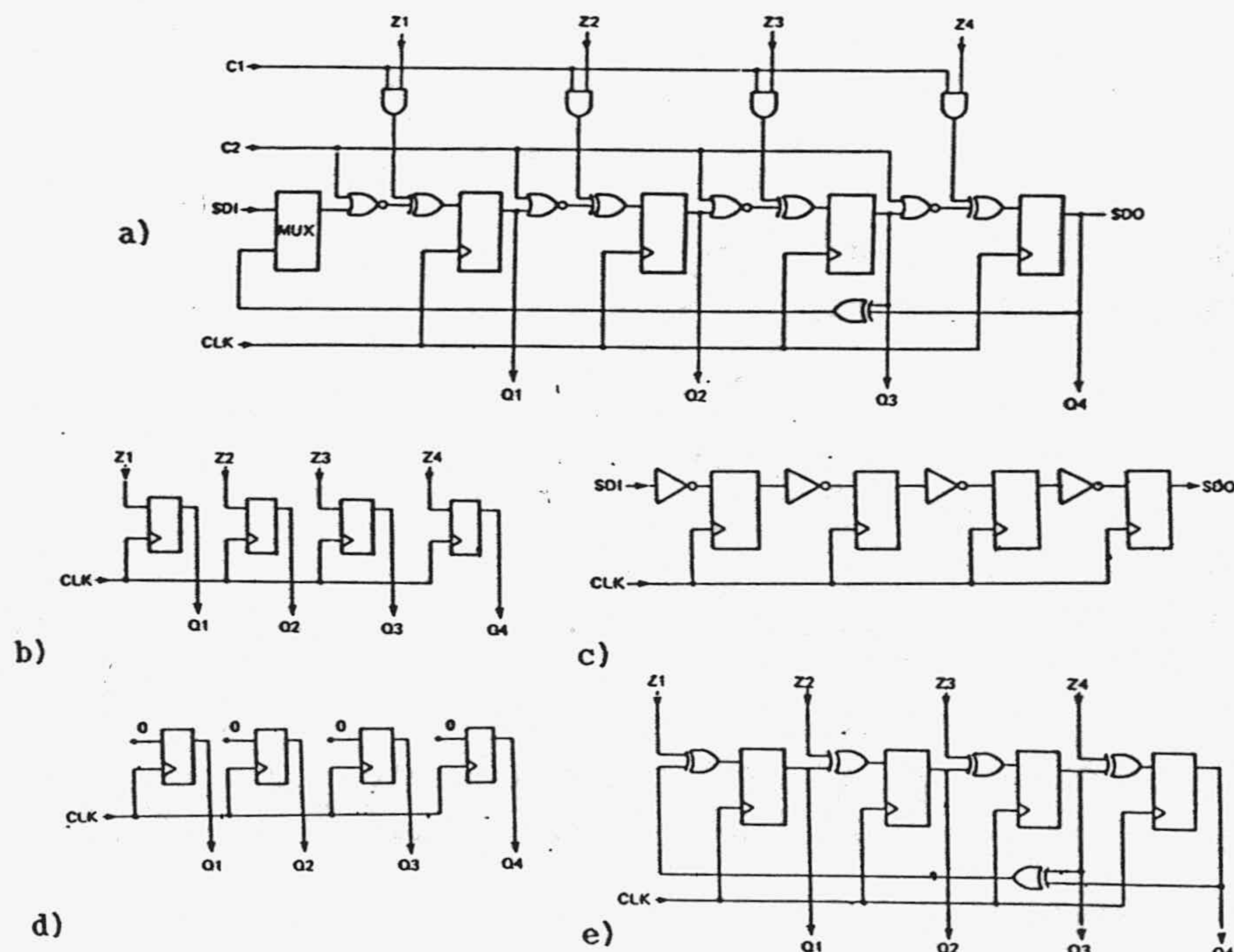


Figura 1.23. Funcionamiento de un BILBO: a) aspecto general; b) $C1C2=11$, modo normal de latch; c) $C1C2=00$, modo de registro de desplazamiento; d) $C1C2=01$, iniciación, e) $C1C2=10$, LFSR.

1.6. Las soluciones para el test de circuitos secuenciales.

Para realizar el test en este caso, habrá que llevar el circuito a un estado determinado y una vez ahí aplicar el patrón de test o la secuencia de patrones de test. Por tanto, elegir un patrón de test significa seleccionar una determinada combinación de valores en las entradas y en los biestables del circuito y, además, encontrar la secuencia de entradas que lleven el circuito al estado deseado. Eso significa aumentar enormemente el tiempo necesario para la generación de patrones, además de incrementar el número de entradas a aplicar y con ello crece el tiempo de test. Existen algoritmos que permiten la generación de patrones para estos circuitos [THO71], y programas como LASAR [WAL85] y SOFTG [SNE77].

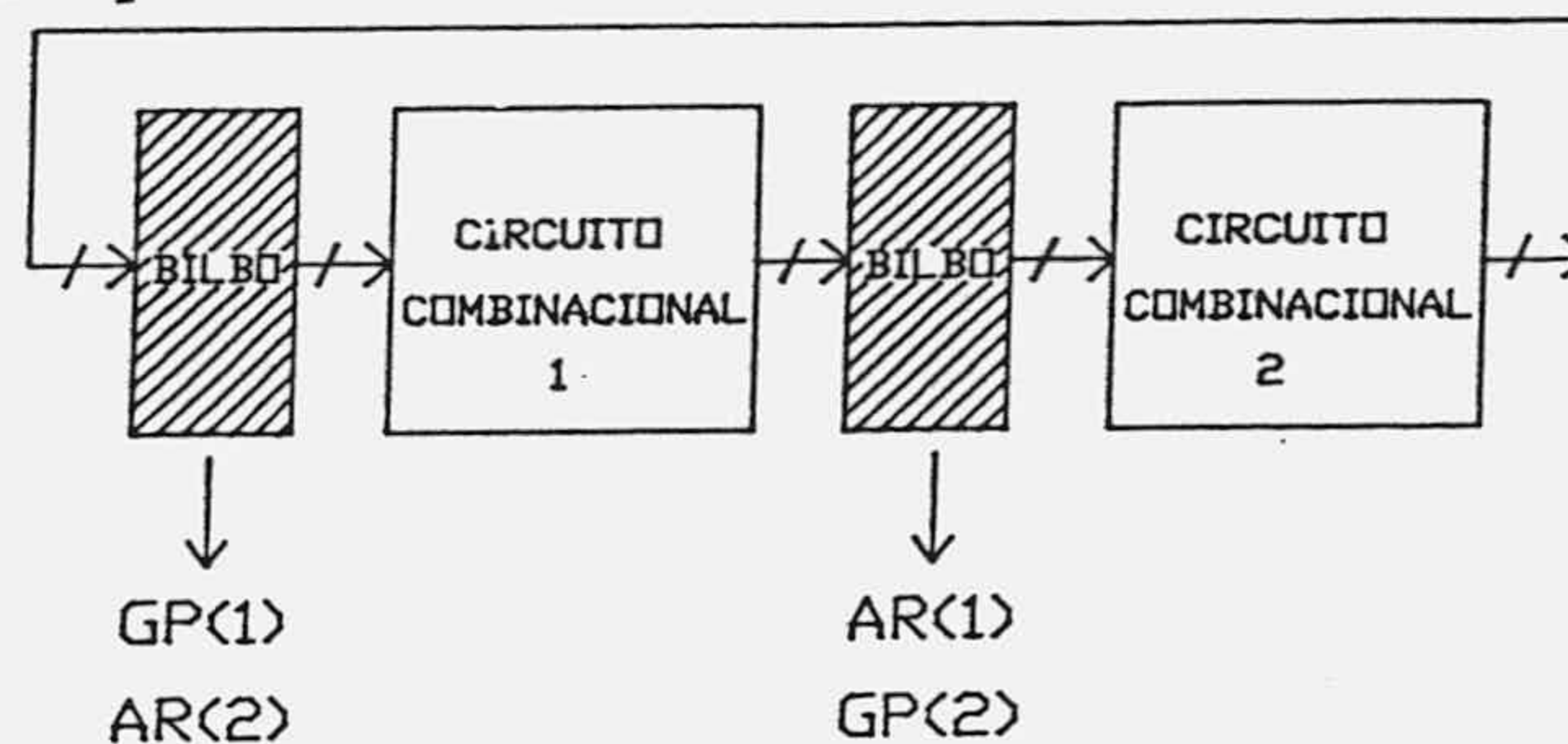


Figura 1.24. Disposición de BILBOs para el test de dos circuitos combinacionales. El primero es GP para el circuito 1 y AR para el circuito 2 y el segundo BILBO es GP para el circuito 2 y AR para el circuito 1.

También, como se ha indicado en la sección 1.3, se puede evitar un tiempo de generación de patrones excesivo mediante los procedimientos de Scan Path [AND80, DAS80, 82, EIC77, FUN78, WIL73]. En ese caso, hay que considerar el incremento en la complejidad del circuito y el tiempo extra necesario para actualizar el valor de los biestables.

1.7. Las soluciones al problema de los defectos que cambian la naturaleza combinacional del circuito.

En este caso, para detectar un defecto habría que aplicar, no ya un patrón de test, sino una secuencia concreta de patrones de test. Eso implicaría modelar esos defectos convenientemente y modificar los simuladores y los algoritmos de búsqueda de patrones de test. Se han desarrollado algunos algoritmos y herramientas que consideran estos posibles defectos, entre ellas cabe citar el simulador MOTIS [L087].

Otra posible solución es la de utilizar procedimientos de test con-

corriente [FUC87, FUJ84, 87, KHA82, SAL88, SAY86, SHI84]. Estos procedimientos comprueban el circuito mientras se encuentra funcionando normalmente, comprobando que la salida obtenida en cada momento o una función de la misma se corresponde con la del circuito correcto. Habrá que añadir una serie de módulos que difieren según el circuito, y es preciso evaluar [BLO90, FUC87, FUJ87] el tiempo necesario para su diseño, la efectividad en la detección de defectos, el incremento de superficie de silicio que suponen y la posible reducción de la velocidad de funcionamiento.

1.8. La solución a los defectos temporales.

No todos los defectos son permanentes, algunos se presentan ocasionalmente y afectan al circuito durante un cierto intervalo de tiempo. Se trata de los defectos temporales, dentro de los cuales se puede distinguir entre defectos transitorios y defectos intermitentes [LAL85].

Los defectos transitorios son los que se manifiestan sólo durante un cierto intervalo de tiempo. Normalmente, se originan por fluctuaciones en la fuente de alimentación, partículas α , etc. y no suponen la existencia de un defecto físico en el circuito. Los defectos intermitentes afectan al circuito en intervalos discretos de tiempo (en esos intervalos se dice que el defecto está activo) y pueden deberse a algunas deficiencias en el diseño que hacen que en ciertas condiciones de funcionamiento se produzcan errores, defectos en los contactos, etc. Normalmente, el comportamiento de estos defectos se representa mediante modelos probabilísticos de Markov [LAL85]. La detección de estos defectos únicamente se puede asegurar si se utiliza un procedimiento de test concurrente que monitorice continuamente el circuito a testear.

1.9. Las soluciones al problema de la degradación.

Con el tiempo, un circuito puede funcionar de manera incorrecta por el deterioro de sus componentes, que puede ocasionar que defectos intermitentes se hagan permanentes y que aparezcan nuevos defectos permanentes y temporales [LAL85]. Para detectar esta situación, se puede utilizar un procedimiento de test concurrente o bien la aplicación periódica de otro tipo de test. Si se opta por la aplicación periódica de un test, los procedimientos de test BIST resultan especialmente adecuados, ya que el circuito dispone internamente de

los módulos para generar los patrones y analizar las respuestas de forma automática, sin necesidad de colocar el circuito integrado en un sistema de aplicación de test (ATE).

1.10. Clasificación de los procedimientos de test de circuitos integrados.

Existen múltiples estrategias para realizar el test de los circuitos integrados y plasmar las soluciones anteriormente descritas. Aunque algunas de esas estrategias se han apuntado en el apartado 1.2, en éste apartado se expone una taxonomía adecuada para la clasificación de los procedimientos de test aparecidos hasta el momento (figura 1.25).

En primer lugar, se puede distinguir entre **test off-line** y **test on-line** o **concurrente**, según se interrumpa el normal funcionamiento del circuito para aplicar el test o, en cambio, el circuito se testeé mientras funciona.

Tanto el **test off-line** como el **test on-line** se realizan mediante la aplicación de una serie de entradas, los **patrones de test**, para las que el circuito a testear generará las correspondientes salidas. Comparando las salidas obtenidas (o alguna función de las mismas) con las salidas correctas (o función de ellas) se determina si el circuito testeado funciona bien o no. En el caso del **test on-line** el proceso se lleva a cabo mientras que el circuito está funcionando; en cambio el **test off-line** se realiza cuando el circuito pasa a un modo de funcionamiento especial (**modo de test**) o bien aplicando los patrones de test en un entorno distinto de donde luego funcionará (un banco de test).

Se puede establecer una primera clasificación de los procedimientos de test, tanto **off-line** como **on-line**, según la **forma de seleccionar** los patrones de test. Así se tienen:

(a) Procedimientos exhaustivos (EXH), si los patrones de test son todas las posibles combinaciones de entradas al circuito, para todos los posibles estados internos del mismo.

(b) Procedimientos pseudoexhaustivos (PEXH), si el circuito se divide en una serie de módulos independientes y se aplican todas las posibles entradas para todos los posibles estados internos de cada uno de los módulos en que se ha dividido el circuito.

(c) Procedimientos con elección de patrones de test (EPT), si se aplican al circuito a testear un subconjunto de las posibles combinaciones de entradas al circuito que pongan de manifiesto los defectos del mismo.

(d) Procedimientos pseudoaleatorios (PSA), si se aplica un conjunto de patrones de test que se ha seleccionado aleatoriamente.

Dentro de cada uno de los tipos anteriores se puede distinguir entre:
 (α) Procedimientos de aplicación combinacionales, en los que no importa el orden de aplicación de los patrones de test.

(β) Procedimientos de aplicación secuenciales, en los que en la definición del test no sólo se considera el conjunto de patrones de test, sino que además interviene la secuencia de aplicación de los mismos.

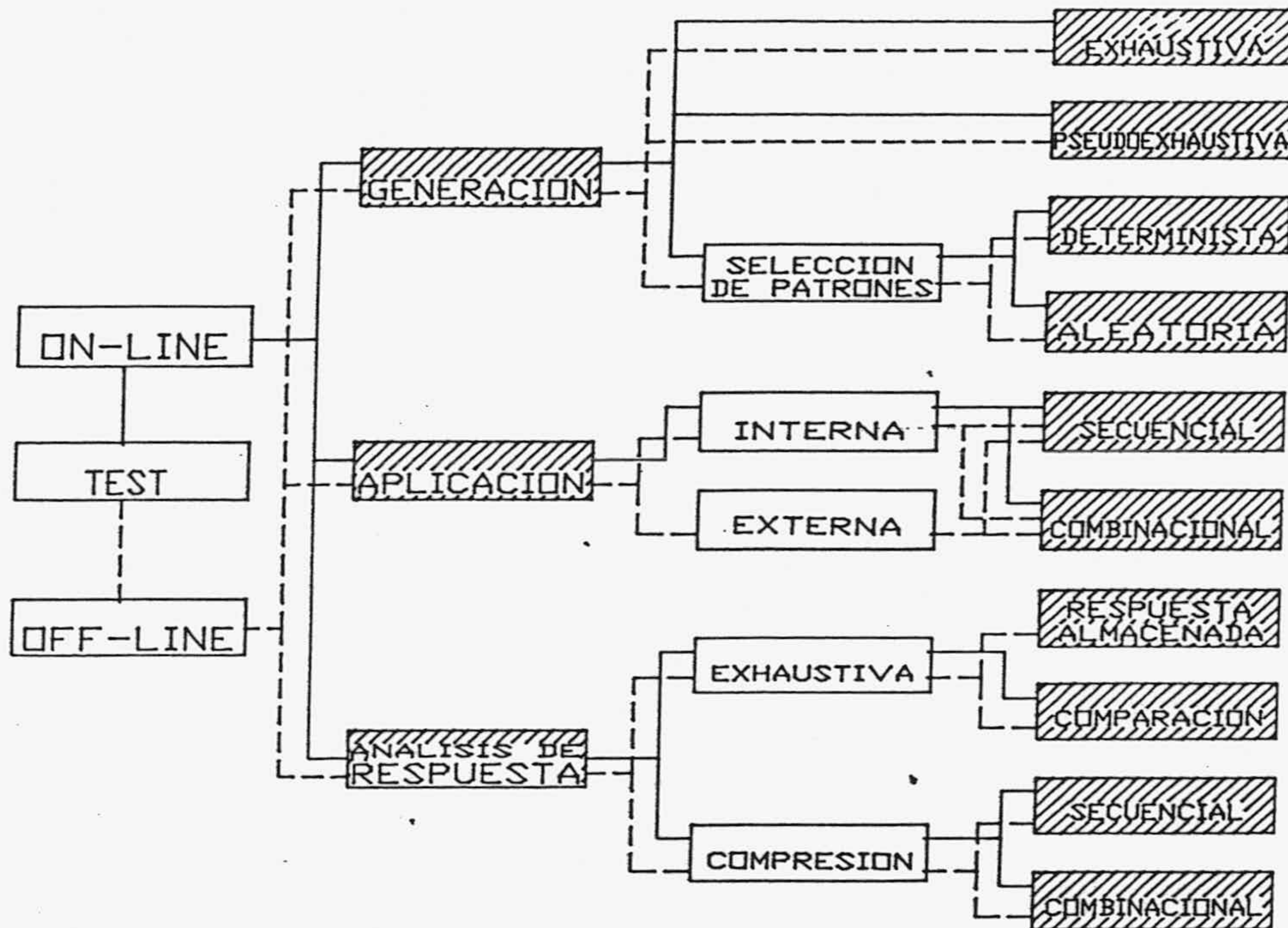


Figura 1.25. Clasificación de los procedimientos de test.

Por otra parte, para completar el test es preciso analizar las respuestas a los patrones de test. Según la forma en que se consideren las respuestas a los patrones de test aplicados se puede distinguir entre:

(i) Procedimientos de análisis exhaustivo de salidas (AEXH), si se analiza la respuesta del circuito a cada patrón de test.

(ii) Procedimientos de compresión o de análisis de firmas (ASG), si se comparan funciones de las salidas del circuito. Dentro de estos procedimientos se encuentran los procedimientos de análisis de firmas combinacionales (ASGC), en los que se analiza una función de cada una de las respuestas a cada patrón, o bien procedimientos de análisis de firmas secuenciales (ASGS), en los que se analiza el valor de una función de la secuencia de respuestas

del circuito a la secuencia de patrones de test aplicada. Normalmente, como la función que se aplica a las respuestas al test implica una disminución del número de bits que se necesita para representarlas, se habla de función **compresión de las salidas** y el resultado recibe el nombre de **signatura del circuito**. Según el procedimiento de compresión por el que se ha obtenido la signatura, se puede distinguir entre **procedimientos de análisis de síndrome, de cuenta de transiciones, con LFSRs, espectrales, cuadráticos, etc.**

En las Tablas 1.2 y 1.3 se muestran las combinaciones de procedimientos de análisis de salidas y de aplicación de patrones de test que tienen sentido para el caso del test off-line y on-line, respectivamente. Esas combinaciones definen los distintos procedimientos de test posibles. Las casillas que no aparecen marcadas corresponden a las combinaciones que no tienen sentido, como ocurre con los procedimientos de análisis de signaturas secuenciales si los patrones se aplican según un procedimiento combinatorial. En la Tabla 1.4 se relacionan los distintos procedimientos de test y los problemas cuya resolución permiten de manera más efectiva.

1.11. Objetivos y contenidos de la memoria.

En este capítulo se han presentado los problemas del test de circuitos integrados y los distintos procedimientos propuestos para superarlos. A modo de resumen de lo expuesto se pueden establecer, como finalidades de los estudios relativos al test de circuitos integrados, las siguientes:

- Tanto para la generación de patrones de test como para la evaluación de procedimientos de selección aleatoria, es esencial disponer de modelos funcionales de los defectos. Es decir, saber qué combinaciones de entradas dan lugar a una salida errónea sin necesidad de recurrir a modelar los defectos a un nivel estructural más complejo, pues dada la cantidad de elementos que incluye un circuito VLSI, puede resultar inviable.

- Dada la poca accesibilidad a los módulos de un circuito integrado, es conveniente incluir la circuitería necesaria para realizar el test junto al módulo que se testea. Esa circuitería, constituida esencialmente por un circuito generador de patrones y un compresor de salidas, ha de ser lo suficientemente simple.

- Se han de desarrollar técnicas que permitan diseñar procedimientos de compresión de respuestas a partir de los requerimientos de cubrimiento que se establezcan.

- Se necesita una mayor investigación para el desarrollo de modelos de error que permitan una evaluación correcta del enmascaramiento, así como técnicas de análisis adecuadas.

TABLA 1.2. Taxonomía de los procedimientos de Test Off-line.

Entradas	Salidas	AEXH	ASGC	ASGS
EXH	Combin.	X	X	
	Secuenc.	X	X	X
PEXH	Combin.	X	X	
	Secuenc.	X	X	X
EPT	Combin.	X	X	
	Secuenc.	X	X	X
PSA	Combin.	X	X	
	Secuenc.	X	X	X

TABLA 1.3. Taxonomía de los procedimientos de Test On-line.

Entradas	Salidas	AEXH	ASGC	ASGS
EXH	Combin.	X	X	
	Secuenc.			
PEXH	Combin.	X	X	
	Secuenc.			
EPT	Combin.	X	X	
	Secuenc.			
PSA	Combin.	X	X	
	Secuenc.			

TABLA 1.4. PROCEDIMIENTOS DE TEST Y PROBLEMAS

Problemas	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)
Procedimientos								
Off-line								
Generacion de Patrones								
EXHs					X	X		
PEXHc	X	X						
PEXHs	X	X			X	X		
SDETC	X							
SDETs	X				X	X		
SALEc	X	X						
SALEs	X	X			X	X		
Análisis de Salidas								
ACSc			X	X				
ACSS			X	X	X	X		X
On-line o Concurrentes								
Generacion de Patrones								
PEXHc	X	X			X	X	X	X
SDETC	X	X			X	X	X	X
SALEc								
Análisis de Salidas								
AEXH			X	X				
ACSc			X	X				

En esta memoria se estudia la utilización de técnicas espectrales en la consecución de los anteriores objetivos. Concretamente se presenta una nueva transformación espectral dotada de un algoritmo de cálculo rápido y aplicable a funciones digitales multivaluadas (con un número primo de valores). En el caso de aplicarse a funciones binarias, la transformada que se obtiene está formada por los coeficientes de la expansión de Reed-Muller de polaridad cero. Su definición, el estudio de sus propiedades y su utilización en el test de circuitos lógicos se realizan en los capítulos 2 y 3 de la presente memoria.

A diferencia de otras herramientas espectrales utilizadas en el estudio de funciones booleanas, la transformada que aquí proponemos es tal que los

coeficientes espectrales que se obtienen a partir de ella están definidos sobre el mismo conjunto que los valores de la función. Por consiguiente, se pueden sintetizar con igual dificultad circuitos que realicen operaciones sobre ambos dominios.

La herramienta espectral y la metodología de realización del test que su uso significa no pretenden ser la respuesta definitiva a los problemas del test de circuitos integrados. En algunos tipos concretos de circuitos puede que esto sea efectivamente así, pero en otros casos sólo contribuyen a reducirlos, incluso puede utilizarse en combinación con algunas otras técnicas y permitir aprovechar ciertas herramientas para nuevos usos. Como hemos visto, el problema de la generación óptima de patrones de tests es un problema NP-completo y por consiguiente, no se puede encontrar una solución de complejidad polinómica para cualquier circuito: hay que buscar la técnica más adecuada según los casos. En este sentido, se necesita todavía un trabajo considerable para establecer métodos de evaluación de procedimientos de test y técnicas de diseño de circuitos testeables. En la actualidad, en la mayoría de los casos se aplican soluciones ad hoc, que la experiencia ha demostrado válidas en muchos casos prácticos. Algunas de las aportaciones de esta memoria se apuntan al desarrollo de estas técnicas de análisis.

Mediante las técnicas espectrales basadas en la transformada de Tamari que se define en el capítulo 2, se ha construido una serie de herramientas y se ha elaborado una serie de procedimientos cuya descripción constituye el objeto de los restantes capítulos. Mediante estas herramientas:

- Se establece una relación entre defectos y cambios en los coeficientes espectrales.
- Se hace una serie de precisiones respecto a cuales son las características que han de tener los circuitos fácilmente testeables mediante estas técnicas espectrales.
- Se define un procedimiento para el test de circuitos complejos a partir del test de los módulos, más simples, que lo componen y para los que se tiene conocimiento de los cambios que los defectos originan en los coeficientes espectrales.
- Se desarrolla un procedimiento para diseñar módulos testeables concurrentemente. El procedimiento permite establecer una relación entre la efectividad de detección de defectos y complejidad del hardware a emplear.

Por último se presentarán ejemplos de aplicación de los procedimientos que se han elaborado.

CAPITULO 2. LA TRANSFORMADA DE TAMARI.

Las técnicas espectrales se han utilizado frecuentemente en el análisis y en la síntesis de funciones discretas [HUR85, KAR76, LEC71]. Mediante el uso de estas técnicas se pueden clasificar las funciones con arreglo a sus propiedades de simetría [EDW75, HUR77, LEC71] de cara a realizar la síntesis modular y el diseño de circuitos testeables [BES78, ERI86, MOR78, MIL81, SOU78, TOK80]. Con ellas se pretende satisfacer la demanda de técnicas de diseño analíticas utilizables en circuitos digitales complejos [KAR85] ya que, como se vió en el capítulo anterior, muchos de los problemas centrales del diseño lógico no son susceptibles de resolverse en la situación más general mediante un algoritmo de complejidad polinómica.

En los últimos años se ha visto incrementado el interés en la aplicación de técnicas espectrales para la detección de defectos en los circuitos lógicos [ERI84, MIL83, MIL84, MOR85, MUZ82, SUS83, TRA85, DAR89a, DAR89b, RUI89]. Los procedimientos de test espectrales se basan en la asignación de un conjunto de coeficientes espectrales a cada una de las funciones que sintetiza un circuito digital, de forma que cuando presente algún defecto se produzca un cambio en alguno de esos coeficientes. El conjunto mínimo de coeficientes espectrales que permita detectar todos los defectos del circuito, o bien, un número suficiente de éstos, constituye la **signatura o firma** del circuito [MIL85, SER86, SUS83].

Los aspectos a considerar cuando se pretende emplear un procedimiento de test espectral son:

a) La complejidad de cálculo que encierra la obtención de los coeficientes espectrales de las funciones del circuito.

b) La facilidad con que se relacionen los posibles defectos del circuito, o más exactamente las faltas que los modelan, con los cambios que originan en los coeficientes espectrales.

c) La posibilidad de seleccionar un conjunto de coeficientes espectrales eficiente a la hora de proporcionar un cubrimiento de faltas aceptable. Por conjunto de coeficientes espectrales eficiente se entiende aquél en el que para calcular sus elementos se necesite el menor número de valores de la función y en el que la circuitería extra debida al módulo de test sea lo más reducida posible.

Hasta el momento la mayoría de los trabajos realizados sobre el tema en circuitos digitales binarios se han centrado en el espectro de Walsh [HUR85, MIL84, 85, MUZ82, SER86, SUS83], ya sea en la versión de Rademacher-Walsh (RW) o en la de Walsh-Hadamard (WH): la diferencia entre ambas versiones se debe a la forma en que se ordenan y designan los coeficientes [HUR85, KAR85]. En el contexto del estudio de las funciones multivaluadas se ha utilizado una generalización de la transformadas de Walsh al caso no binario, la transformada de Chrestenson [HUR85, KAR85, MOR78, MOR85, TOK79]. No obstante, el espectro de Walsh presenta algunas características que pueden suponer ciertos problemas en su utilización como herramienta espectral, de acuerdo con los planteamientos realizados anteriormente [DAR89]:

a) Cada uno de los coeficientes del espectro de Walsh para una función de n variables se obtiene sumando (o restando) el valor de la misma en cada una de las 2^n combinaciones de entradas. Por consiguiente, para valores de n grandes, la obtención del espectro de WH resulta bastante costosa.

b) Los coeficientes del espectro de Walsh son enteros comprendidos entre 2^n y -2^n para una función de n variables binarias. Para codificar cada coeficiente del espectro de Walsh de una función de n variables binarias se pueden precisar n bits, lo cual debe tenerse en cuenta en el caso de que se pretenda integrar el procedimiento de test.

En este capítulo se describe una nueva transformación desarrollada con vistas a su aplicación al test de circuitos lógicos. Se trata de la transformada de Tamari, una transformada que se puede definir para funciones p -valuadas con p primo, y por consiguiente es aplicable en particular a los circuitos binarios. El nombre con el que se ha bautizado, transformada de Tamari, se debe a que Tamari [TAM52] sugiere utilizar las mismas funciones que se han usado aquí en la definición de la nueva transformada, para desarrollar

cualquier función p-valuada, con p primo.

La transformada de Tamari, de forma similar a lo que ocurre con las transformadas de Walsh y de Chrestenson, se puede calcular mediante un procedimiento de cálculo rápido del mismo tipo que la transformada rápida de Fourier (FFT) [KUN80]. En este capítulo también se expondrá dicho procedimiento, desarrollado por nosotros. Además, como se mostrará en el presente capítulo, existen dos aspectos interesantes de cara a su utilización como herramienta en los circuitos digitales y que suponen unas ciertas ventajas respecto a la transformada de Walsh o de Chrestenson:

a) Los coeficientes espectrales de Tamari de una función p-valuada, son también elementos p-valorados.

b) Los coeficientes espectrales no dependen de los valores de la función para todas y cada una de las entradas sino de un conjunto de ellos, específico para cada coeficiente.

En el caso de que p sea igual a 2, la transformada de Tamari de una función permite obtener los coeficientes de Reed-Muller [DAR89a, DAR89b, GRE90] de dicha función, por lo que constituye un procedimiento de cálculo rápido de los mismos.

En un primer apartado se define la transformada de Tamari, para pasar a describir en el siguiente el algoritmo rápido de cálculo o transformada rápida de Tamari (FTT). Finalmente se muestra la utilización de esta transformada en el test de circuitos digitales p-valorados, estableciendo la forma de relacionar las faltas que modelan los defectos del circuito con los cambios que se producen en los coeficientes espectrales.

2.1. La Transformada de Tamari.

En este primer apartado se presenta la transformada de Tamari [TAM52], cuyo estudio y aplicación al test de circuitos lógicos constituye el objetivo central de la presente memoria.

El conjunto $C_p = \{0, 1, 2, \dots, p-1\}$, p primo, con las operaciones suma módulo-p (\oplus) y producto módulo-p (\cdot), constituye un Cuerpo Finito o **Cuerpo de Galois** [KNU85]. El producto cartesiano de C_p con él mismo m veces, define el conjunto C_p^m , cuyos elementos se notarán con letras mayúsculas, de forma que Y , $Y = \sum_{i=0}^{m-1} y_{i+1} p^i$, designa al elemento $(y_1, \dots, y_m) \in C_p^m$ (con $y_i \in C_p$).

Sea el conjunto de las funciones de C_p^m en C_p , $F_p^m = \{f/f: C_p^m \rightarrow C_p\}$, en el que se definen los siguiente operadores:

Definición 2.1 (Suma módulo-p, #): Dadas dos funciones $f, g \in F_p^m$, la función suma módulo-p de esas dos funciones, $h=f\#g$, se define como

$$h(Y) = f(Y) \oplus g(Y), \quad \forall Y \in C_p^m.$$

Definición 2.2 (Producto escalar módulo-p, *): Dado un elemento $a \in C_p$, y una función $f \in F_p^m$, la función producto escalar módulo-p de a por f, $g=a*f$, se define como

$$g(Y) = a \cdot f(Y), \quad \forall Y \in C_p^m$$

Con estas dos operaciones definidas en el conjunto F_p^m se tiene que $V(F_p^m, C_p, \#, *)$ es un espacio vectorial de dimensión p^m . En ese espacio vectorial se puede definir una base constituida por p^m funciones de F_p^m , B_0, B_1, \dots, B_N (con $N=p^m-1$) de manera que cualquier función de F_p^m se puede expresar de forma unívoca como combinación lineal de la funciones de la base:

$$f = (r_0 * B_0) \# \dots \# (r_N * B_N) = \sum_i r_i * B_i, \quad r_i \in C_p \quad (i=0, \dots, N) \quad (2.1)$$

donde se ha notado con Σ' la suma módulo-p, #, definida en F_p^m . De esta forma, cada combinación de coeficientes r_0, r_1, \dots, r_N define una función de F_p^m de manera unívoca y recibe el nombre de **espectro de la función** en la base seleccionada. A continuación, se considera el conjunto de funciones de F_p^m a partir de las que se construirá la Transformada de Tamari.

El conjunto de funciones de F_p^m , T_i ($i=0, \dots, N$):

$$T_i(Y) = y_1^{i_1} \cdot y_2^{i_2} \cdot \dots \cdot y_m^{i_m} \quad (2.2)$$

$$\begin{aligned} \text{donde} \quad i &= i_1 + i_2 p^1 + \dots + i_m p^{m-1}, \quad i_j \in C_p \\ Y &= y_1 + y_2 p^1 + \dots + y_m p^{m-1}, \quad y_j \in C_p \\ Y^1 &= Y \cdot Y \cdot \dots \cdot Y \\ 0^0 &= 1^0 = 1, \end{aligned}$$

constituyen una base del espacio vectorial $V(F_p^m, C_p, \#, *)$ denominada base de Tamari, BT, [HUR77]. Cada función $T_i \in BT$, $i=0, \dots, N$, se representa mediante un vector de $N+1$ componentes correspondientes a los valores de la función para cada una de las $N+1$ combinaciones de entradas:

$$[T_i(Y)] \equiv [t(i,0) \quad t(i,1) \quad \dots \quad t(i,N)] \quad (2.3)$$

$$\begin{aligned} \text{con} \quad t(i,j) &= T_i(j), \quad i=0, \dots, N, \quad T_i(Y) \in C_p \\ j &= y_1 + y_2 p + \dots + y_m p^{m-1} \end{aligned}$$

que se utilizan para construir una matriz de $p^m \times p^m$ elementos:

$$M(p,m) = [[T_0(Y)] [T_1(Y)] \dots [T_N(Y)]]^T \quad (2.4)$$

Como ejemplo, la base de Tamari para las funciones de F_3^2 está constituida por 9 funciones cuyos vectores asociados son:

$$\begin{aligned} T_0 &= [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \\ T_1 &= [0 \ 1 \ 2 \ 0 \ 1 \ 2 \ 0 \ 1 \ 2] \\ T_2 &= [0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1] \\ T_3 &= [0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 2 \ 2 \ 2] \\ T_4 &= [0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 0 \ 2 \ 1] \\ T_5 &= [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 2 \ 2] \\ T_6 &= [0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \\ T_7 &= [0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 0 \ 1 \ 2] \\ T_8 &= [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1] \end{aligned}$$

y la matriz $M(3,2)$ asociada es

$$M_3^2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & 2 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Igualmente, cualquier otra función, $f \in F_p^m$, se puede representar mediante un vector de $N+1$ componentes construido de la misma forma que en (2.3):

$$[f] \equiv [f(0) \ f(1) \ \dots \ f(N)], \quad f(i) \in C_p \quad (2.5)$$

Definición 2.3 (Producto de una función de F_p^m por una matriz): Dada la matriz $[A]=[a(i,j)]$, con $a(i,j) \in C_p$ y $j,i=0,\dots,N$, y la función $f \in F_p^m$ representada por el vector $[f(0) \ f(1) \ \dots \ f(N)]$, el producto de f por $[A]$ es una función $g \in F_p^m$ representada por el vector $[g]=[g(0) \ g(1) \ \dots \ g(N)]$ cuyas componentes se definen como sigue:

$$\begin{aligned} g(i) &= (f(0) \cdot a(0,i)) \oplus (f(1) \cdot a(1,i)) \oplus \dots \oplus (f(N) \cdot a(N,i)) = \\ &= \sum_j f(j) \cdot a(j,i) \quad (j=0,1,\dots,N) \end{aligned} \quad (2.6)$$

El producto de una función $f \in F_p^m$ por la matriz $[A]$ de $p^m \times p^m$ elementos $a(j,i) \in C_p$ se notará como

$$[g]=[f] \cdot [A] \quad (2.7)$$

Puesto que las funciones definidas en (2.2) constituyen una base,

cualquier función $f \in F_p^m$ se puede expresar de manera unívoca como combinación lineal de ellas de forma que:

$$f(Y) = (r_0 * T_0(Y)) \# (r_1 * T_1(Y)) \# \dots \# (r_N * T_N(Y)), \quad \forall Y \in C_p^m \quad (2.8)$$

Es decir, que

$$f(Y) = (r_0 * t(0, Y)) \# (r_1 * t(1, Y)) \# \dots \# (r_N * t(N, Y)), \quad \forall Y \in C_p^m \quad (2.9)$$

y teniendo en cuenta (2.6):

$$[f] = [r_0 \ r_1 \ \dots \ r_N] \cdot [[T_0(Y)] \ [T_1(Y)] \ \dots \ [T_N(Y)]]^T \quad (2.10)$$

donde el vector de componentes r_i , $i=0,1,\dots,N$, define el espectro de Tamari de la función $f \in F_p^m$. Se notará $[f^*]$.

Obsérvese que los elementos del espectro pertenecen a C_p . Consiguientemente, el espectro de Tamari de una función $f \in F_p$ define otra función $f^* \in F_p$ tal que $f^*(i) = r_i$ para todo $i=0,\dots,N$.

A partir de (2.10) se tiene que $[f] = [f^*] \cdot M(p, m)$, y por consiguiente, para obtener el espectro de Tamari de una función $f \in F_p^m$ dada, hay que calcular la matriz inversa de $M(p, m)$.

La existencia de $M(p, m)^{-1}$ está asegurada al construirse las filas de $M(p, m)$ con los vectores asociados a las funciones de Tamari, T_i , $i=0,\dots,N$, que son linealmente independientes. Así, notando $T(p, m) = M(p, m)^{-1}$, queda:

$$[f^*] = [f] \cdot T(p, m) \quad (2.11)$$

siendo $T(p, m)$ la matriz de la transformada de Tamari sobre F_p^m .

En el caso de F_3^2 queda:

$$T(3, 2) = \begin{bmatrix} 1 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 2 & 2 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 2 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 & 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 2 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 1 \end{bmatrix}$$

En resumen, para calcular el espectro de Tamari $[f^*]$ de una función $f \in F_p^m$ hay que completar los siguientes pasos:

1. Obtener las p^m funciones de la base de Tamari correspondiente a F_p^m y construir la matriz $M(p,m)$.
2. Calcular $T(p,m)=M(p,m)^{-1}$.
3. Realizar el producto $[f].T(p,m)$.

Este procedimiento implica un volumen de cálculo que depende exponencialmente de m y resulta inviable para valores elevados de m . En la siguiente sección se resuelve esta dificultad mediante la construcción de un algoritmo de cálculo rápido de la transformada: la **Transformada Rápida de Tamari (Fast Tamari Transform, FTT)**. No obstante, antes de pasar al siguiente apartado se demuestra la siguiente propiedad de la transformada de Tamari:

Propiedad 2.1. (Linealidad): Dadas las funciones $f, g \in F_p^m$, con espectros de Tamari $[f^*]$ y $[g^*]$, respectivamente, el espectro de la función $h \in F_p^m$, siendo $h = \alpha * f \# \beta * g$, verifica que $[h^*] = \alpha * [f^*] \# \beta * [g^*]$.

Demostración: Utilizando (2.9) se tiene que

$$f(Y) = (r_0 * t(0, Y)) \# (r_1 * t(1, Y)) \# \dots \# (r_N * t(N, Y)),$$

$$\text{y } g(Y) = (s_0 * t(0, Y)) \# (s_1 * t(1, Y)) \# \dots \# (s_N * t(N, Y)), \quad \forall Y \in C_p^m.$$

por lo que, utilizando las propiedades de espacio vectorial se obtiene:

$$h(Y) = (w_0 * t(0, Y)) \# (w_1 * t(1, Y)) \# \dots \# (w_N * t(N, Y)), \quad \forall Y \in C_p^m$$

verificándose $w_j = \alpha r_j \oplus \beta s_j, \quad \forall j=0, 1, \dots, N$, como se quería demostrar. ■

2.2. Un algoritmo rápido para la Transformada de Tamari.

En este apartado se presenta un algoritmo rápido para el cálculo de la transformada de Tamari. El algoritmo se denomina **Transformada Rápida de Tamari (FTT)** y se basa en la estructura de la matriz de la transformación $T(p,m)$, que posibilita la aplicación del teorema de Good [KUN80]. Se comprueba que la complejidad de cálculo de la FTT es similar al de las transformadas rápidas de Fourier (FFT), Walsh (FWT) o Chrestenson (FCT) [BES85, KAR85, KUN80], lo cual es lógico si se tiene en cuenta que la obtención de esos algoritmos rápidos también se basan en la aplicación del teorema de Good. En primer lugar, se demuestra el siguiente teorema:

Teorema 2.1: La matriz de la Transformada de Tamari, $T(p,m)$, de una función de F_p^m se puede obtener mediante producto de Kronecker [HUR85] de la matriz $T(p,1)$ por sí misma m veces:

$$T(p,m) = T(p,1) \otimes \dots \otimes T(p,1) \quad (2.12)$$

Demostración: Puesto que $T(p,m)=M(p,m)^{-1}$ y el producto de Kronecker preserva la inversión de matrices, demostrar (2.12) es equivalente a demostrar

$$M(p,m)=M(p,1)\otimes\dots^m\dots\otimes M(p,1) \tag{2.13}$$

Considérese $m=1$, la matriz $M(p,1)$ se definió (2.4) como

$$M(p,1)=[[T_0(Y)] \quad [T_1(Y)] \quad \dots \quad [T_{p-1}(Y)]]^T \tag{2.14}$$

con $T_i \in F_p^{-1}$, $(i=0,1,\dots,p-1)$, $Y \in C_p$

Así, para cualesquiera $j,k \in C_p^{-1}$

$$\begin{aligned} T_j(y_1) &= y_1^j, \quad T_k(y_2) = y_2^k \text{ con } y_1, y_2 \in C_p; \\ T_j(y_1) \otimes T_k(y_2) &= y_1^j \cdot y_2^k = T_r(y_1, y_2) \tag{2.15} \\ &\text{con } r=j+kp \text{ (recuérdese que } p \text{ es primo)} \\ &\text{y } T_r \in F_p^{-2} \end{aligned}$$

Para $M(p,2)$ se tiene que

$$M(p,2)=[[T_0(y_1, y_2)] \quad [T_1(y_1, y_2)] \quad \dots \quad [T_N(y_1, y_2)]]^T \tag{2.16}$$

con $N=p^2-1$. Usando (2.15), de (2.16) se obtiene

$$\begin{aligned} M(p,2) &= [[T_0(y_1) \otimes T_0(y_2)] \dots [T_0(y_1) \otimes T_{p-1}(y_2)] \\ &\dots \dots \dots \\ &\quad [T_{p-1}(y_1) \otimes T_0(y_2)] \dots [T_{p-1}(y_1) \otimes T_{p-1}(y_2)]]^T = \\ &= [[T_0(y_1)] \dots [T_{p-1}(y_1)]]^T \otimes [[T_0(y_2)] \dots [T_{p-1}(y_2)]]^T = \\ &= M(p,1) \otimes M(p,1) \tag{2.17} \end{aligned}$$

Por último, teniendo en cuenta la asociatividad del producto de Kronecker y (2.17)

$$\begin{aligned} M(p,m) &= M(p,m-1) \otimes M(p,1) = M(p,m-2) \otimes M(p,1) \otimes M(p,1) = \\ &= M(p,1) \otimes \dots^m \dots \otimes M(p,1) \quad \blacksquare \end{aligned}$$

Una vez demostrada (2.12) se puede utilizar el teorema de Good para obtener un algoritmo de cálculo rápido. El teorema de Good establece que si una matriz $B(p,m)$ de $p^m \times p^m$ elementos se puede generar mediante m productos de Kronecker de una matriz, $A(p)$, de $p \times p$ elementos, entonces se tiene que

$$B(p,m)=G(p,m)\dots^m\dots G(p,m) \tag{2.18}$$

en donde $G(p,m)$ es una matriz con $p^m \times p^m$ elementos de C_p y que se obtiene a partir de $A(p)$ según la forma dada en [KUN80]:

$$g(i,j) = a(i_m, j_1) \{ \delta(i_m, j_{m-1}) \dots \delta(i_2, j_1) \delta(i_1, j_2) \}$$

$$\begin{aligned} \text{con} \quad & i = i_1 + i_2 p + i_3 p^2 + \dots + i_m p^{m-1}; \\ & j = j_1 + j_2 p + j_3 p^2 + \dots + j_m p^{m-1}; \\ & g(i,j) \text{ es un elemento de } G(p,m), \\ & i, j = 0, 1, \dots, p^m - 1; \\ & a(i_s, j_k) \text{ es un elemento de } A(p), \\ & i_s, j_k = 0, 1, \dots, p-1; \end{aligned} \tag{2.19}$$

En el caso de la transformada de Tamari, para obtener $T(p,m)$ la matriz $A(p)$ es precisamente, $T(p,1) = M(p,1)^{-1}$. Algunas matrices $T(p,1)$ para distintos valores de p son:

$$T(2,1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad T(3,1) = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 2 & 2 \\ 0 & 1 & 2 \end{bmatrix} \quad T(5,1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 4 \\ 0 & 4 & 4 & 4 & 4 \\ 0 & 2 & 1 & 3 & 4 \\ 0 & 3 & 1 & 2 & 4 \\ 0 & 1 & 4 & 1 & 4 \end{bmatrix}$$

Así pues, el algoritmo de cálculo de la transformada de Tamari de una función $f \in F_p^m$ está definido por los siguientes pasos:

1. Obtener las p funciones de la base de Tamari para $V(F_p^1, C_p, \#, *)$, contruir con ellas la matriz $M(p,1)$ y calcular su inverso, es decir, la matriz $T(p,1)$.
2. Construir la matriz $G(p,m)$ a partir de $T(p,1)$ y multiplicar esa matriz m veces por sí misma para obtener $T(p,m)$.
3. Calcular $[f^*] = [f] \cdot T(p,m)$

Como ejemplo, se muestran las matrices de Good $G(3,2)$, para las funciones de F_3^2 y $G(2,3)$, para las de F_2^3 :

$$G(3,2) = \begin{bmatrix} 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{bmatrix} \quad G(2,3) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A partir de ellas se pueden obtener las matrices de la transformación de Tamari:

$$T(3,2) = T(3,1) \otimes T(3,1) = G(3,2)G(3,2)$$

$$T(2,3) = T(2,1) \otimes T(2,1) \otimes T(2,1) = G(2,3)G(2,3)G(2,3)$$

2.2.1. Síntesis utilizando la Transformada de Tamari.

Los coeficientes del espectro de Tamari de una función de F_p^m describen una forma de sintetizar dicha función en términos de puertas lógicas que realicen la suma módulo p y el producto módulo p . En este apartado se ilustra esta utilidad de la Transformada de Tamari a través de un ejemplo. Así, se considera una función de F_3^2 :

$$[g(y_1, y_2)] = [0 \ 1 \ 1 \ 0 \ 1 \ 2 \ 0 \ 0 \ 0]$$

Multiplicando este vector por la matriz $T(3,2)$ de la transformación se obtiene:

$$[0 \ 1 \ 1 \ 0 \ 1 \ 2 \ 0 \ 0 \ 0] \begin{bmatrix} 1 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 2 & 2 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 2 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 & 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 2 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 1 \end{bmatrix} = [0 \ 0 \ 1 \ 0 \ 2 \ 0 \ 0 \ 2 \ 2]$$

Y, consecuentemente la función $g(y_1, y_2)$ se puede expresar utilizando como operadores la suma módulo-3 (\oplus) y el producto módulo-3 (\otimes) en la forma:

$$g(y_1, y_2) = y_1^2 \oplus (y_1 \cdot y_2) \oplus (y_1^2 \cdot y_2^2) \otimes (y_1 \cdot y_2^2)$$

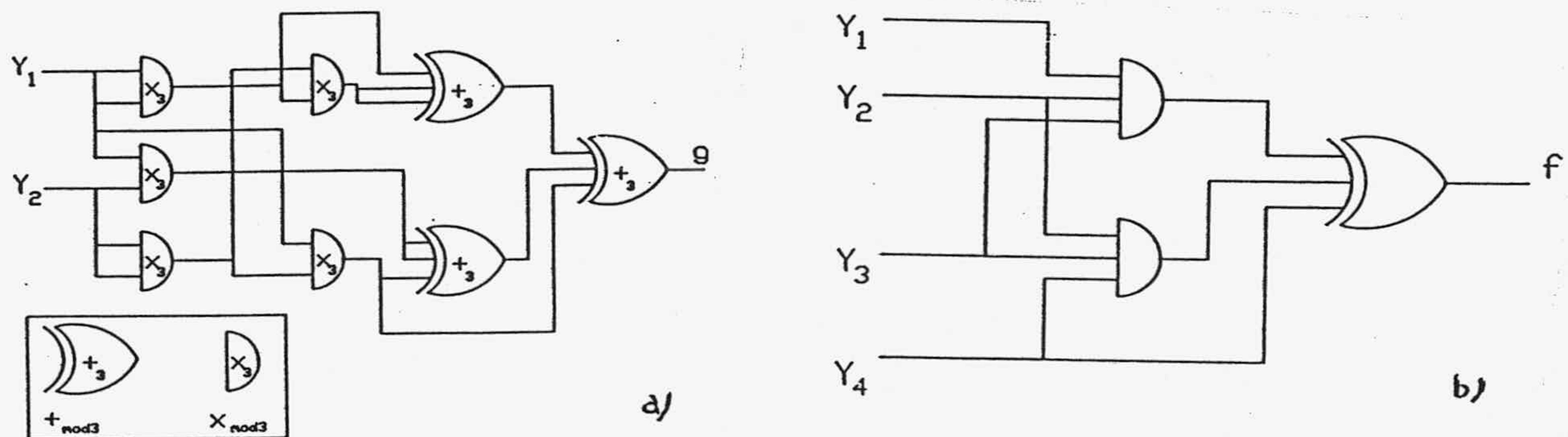


Figura 2.1. Circuitos ejemplo: a) ternario; b) binario.

Así, la transformada de Tamari de una función permite de forma directa la síntesis de la función de F_p^m mediante una red de dos niveles de puertas. En el primer nivel se tendrían las puertas correspondientes al operador producto módulo- p y en el segundo nivel las puertas suma módulo- p . Además

todas las reconvergencias (sección 1.2.6) que existan en el circuito tendrán la misma paridad. Conviene resaltar el hecho de que si $p=2$, las puertas que se utilizan son la EXOR y la AND.

En la figura 2.1.a y 2.1.b se muestran las redes que sintetizan dos funciones: la función $g(y_1, y_2)$ de F_3^2 considerada anteriormente y la función de F_2^4 $[f(Y)] = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1]$, cuyo espectro de Tamari es $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0]$.

2.2.2. Representación gráfica del cálculo de la FTT.

En este apartado se presenta un procedimiento gráfico para la descripción del proceso de cálculo de la transformada basado en la regularidad de las matrices de Good.

Teniendo en cuenta la forma de las matrices de Good, $G(p, m)$, que se utilizan para obtener la FTT en F_p^m , se puede describir de manera simple el procedimiento de cálculo mediante la utilización de grafos de flujo.

Dada una función $f \in F_p^m$, se definen los vectores

$$[r_0^j\ r_1^j\ \dots\ r_N^j],\ N=p^m-1,\ r_i^j \in C_p,\ j=0,1,\dots,m \quad (2.20)$$

de forma que que verifiquen

$$\begin{aligned} \text{i)}\quad & r_i^0 = f(i) \\ \text{ii)}\quad & [r_0^j\ r_1^j\ \dots\ r_N^j] = [r_0^{j-1}\ r_1^{j-1}\ \dots\ r_N^{j-1}]G(p, m) \quad \text{para } j=1,2,\dots,m \end{aligned} \quad (2.21)$$

Así, como $T(p, m) = (G(p, m))^m$, el vector $[r_0^m\ r_1^m\ \dots\ r_N^m]$ corresponde precisamente el espectro de Tamari $[f^*]$ de la función $f \in F_p^m$. Esto significa que el cálculo de la Transformada de Tamari en F_p^m se lleva a cabo mediante m pasos, dependiendo el resultado de cada uno de ellos únicamente del resultado del paso anterior. Esa dependencia está definida por la matriz $G(p, m)$ que se obtiene de $T(p, 1)$ según (2.19). El algoritmo es eficiente en cuanto al uso de memoria de ordenador, ya que sólo hay que memorizar los $N+1$ elementos r_i ($i=0, \dots, N$) de cada paso y los p^2 elementos de $T(p, 1)$.

En la figura 2.2 se muestran los grafos de cálculo para $T(3, 1)$ y $T(2, 1)$. En ellos se ha seguido la notación

$$\begin{aligned} k, i &= k_m p^{m-1} + k_{m-1} p^{m-2} + \dots + k_2 p + i p^0 \\ i, k &= i p^{m-1} + k_m p^{m-2} + k_{m-1} p^{m-3} + \dots + k_2 p^0 \\ \text{con } i, k_j &= 0, 1, \dots, p-1 \end{aligned}$$

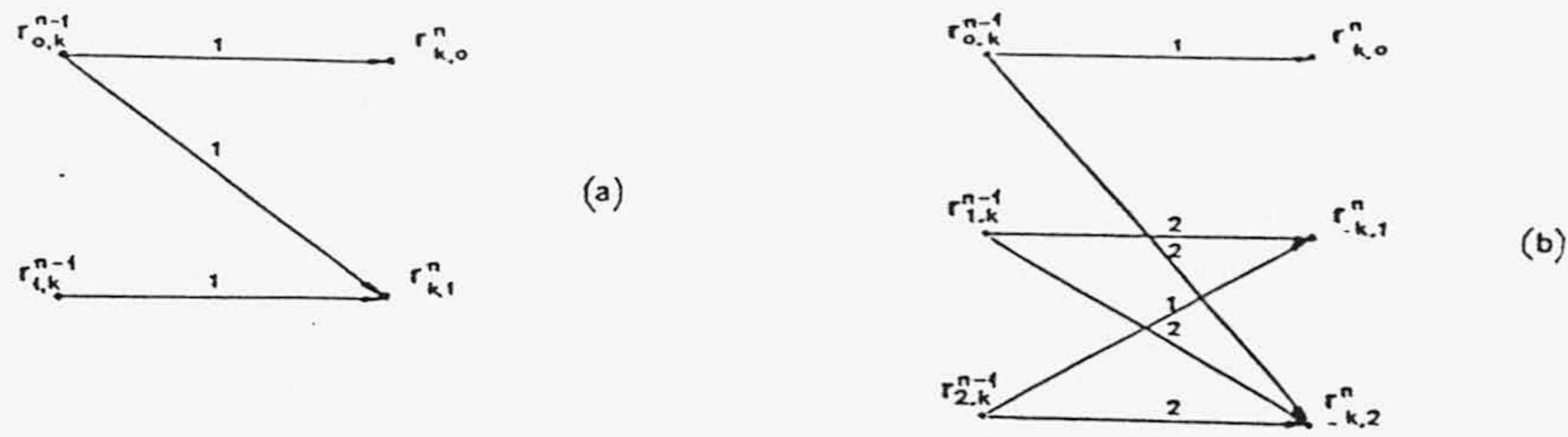


Figura 2.2. Grafos de cálculo para (a) $T(2,1)$, y (b) $T(3,1)$.

En la figura 2.3 se representan los grafos de cálculo de la FFT para las funciones de F_2^3 y F_3^2 .

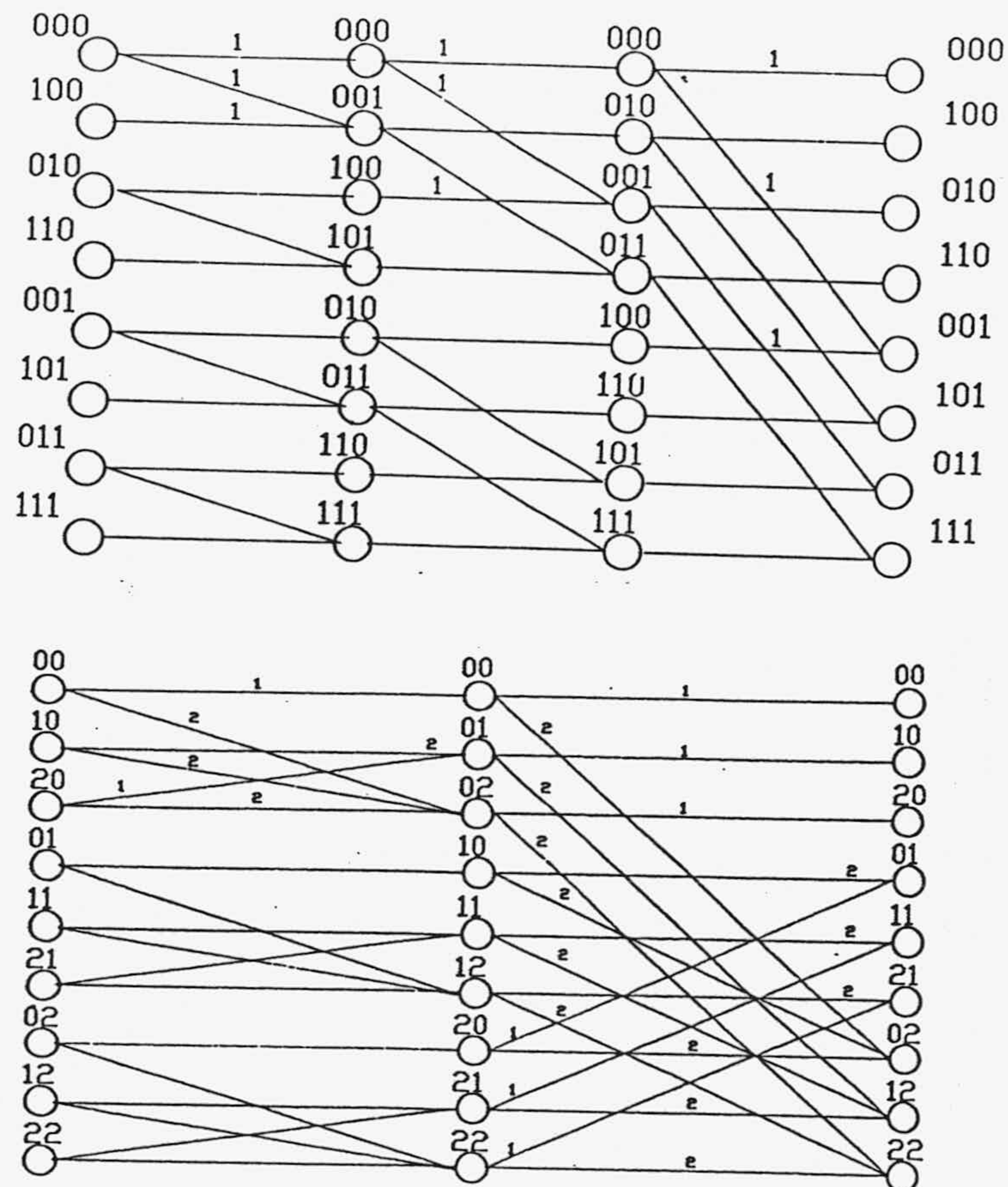


Figura 2.3. Grafos de cálculo para la FFT de F_2^3 y F_3^2

Puesto que la matriz de la transformación $T(p,m)$ se obtiene a partir de m veces el producto de Kronecker de $T(p,1)$ por sí misma, la síntesis de un

circuito para calcular la transformación de Tamari sobre F_p^m se puede hacer de forma modular, utilizando bloques, notados como $P(A_p)$, asociados a la matriz $T(p,1)$. En la red que implementa $T(p,m)$, el número de bloques de cálculo de $T(p,1)$ necesarios es $m \cdot p^{(m-1)}$. Como ejemplos, en las figuras 2.4 y 2.5 se muestran las redes para el cálculo de la transformada de Tamari de las funciones de F_2^3 y F_3^2 , respectivamente.

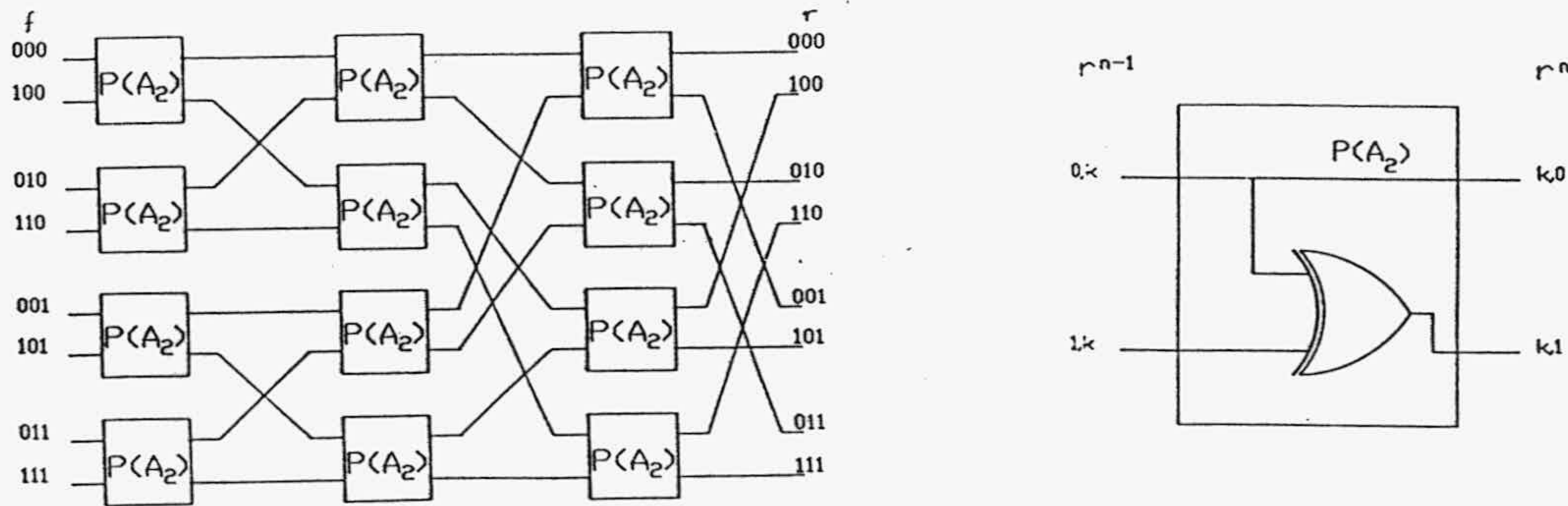


Figura 2.4. Red para la FFT en F_2^3 .

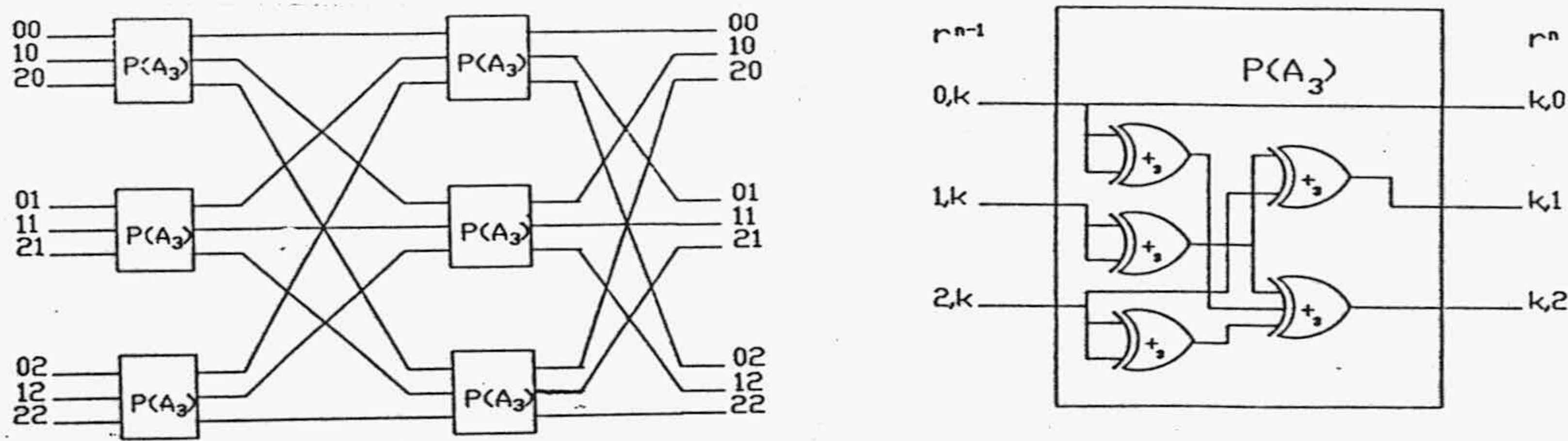


Figura 2.5. Red para la FFT en F_3^2 .

Además, resulta sencillo elaborar un algoritmo para el cálculo de la transformada de Tamari sobre F_p^m en un computador SIMD con una estructura como la que se esquematiza en la figura 2.6 [HWA85]. En ella, cada elemento de proceso (EP) llevaría a cabo el producto de la matriz $T(p,1)$ sobre los p elementos correspondientes y la conexión entre los EPs y los bloques de memoria (MB) se lleva a cabo a través de la red de alineamiento (AN). El algoritmo se ejecuta en m pasos. En cada uno de ellos las p entradas a cada uno de los EPs se toman de los diferentes módulos de memoria, controlándose la red AN para que se lleven a cabo las conexiones adecuadas entre EPs y MBs. Por ejemplo, se supone un SIMD con $N+1$ bloques de memoria, MB_0, MB_1, \dots, MB_N , $s=p^{m-1}$ elementos de proceso, $EP_0, EP_1, \dots, EP_{s-1}$ y los valores de la función a

transformar, r_k^0 ($k=0,1,\dots,N$) almacenados en los bloques MB_k ($k=0,1,\dots,N$). En este caso, en el paso i -ésimo, el elemento de proceso EP_j tomará sus entradas de los módulos de memoria $r, r+1p^{i-1}, r+2p^{i-1}, \dots, r+(p-1)p^{i-1}$, con $r=[j/p^i]p^i+j$ ($[x]$ es la parte entera de x).

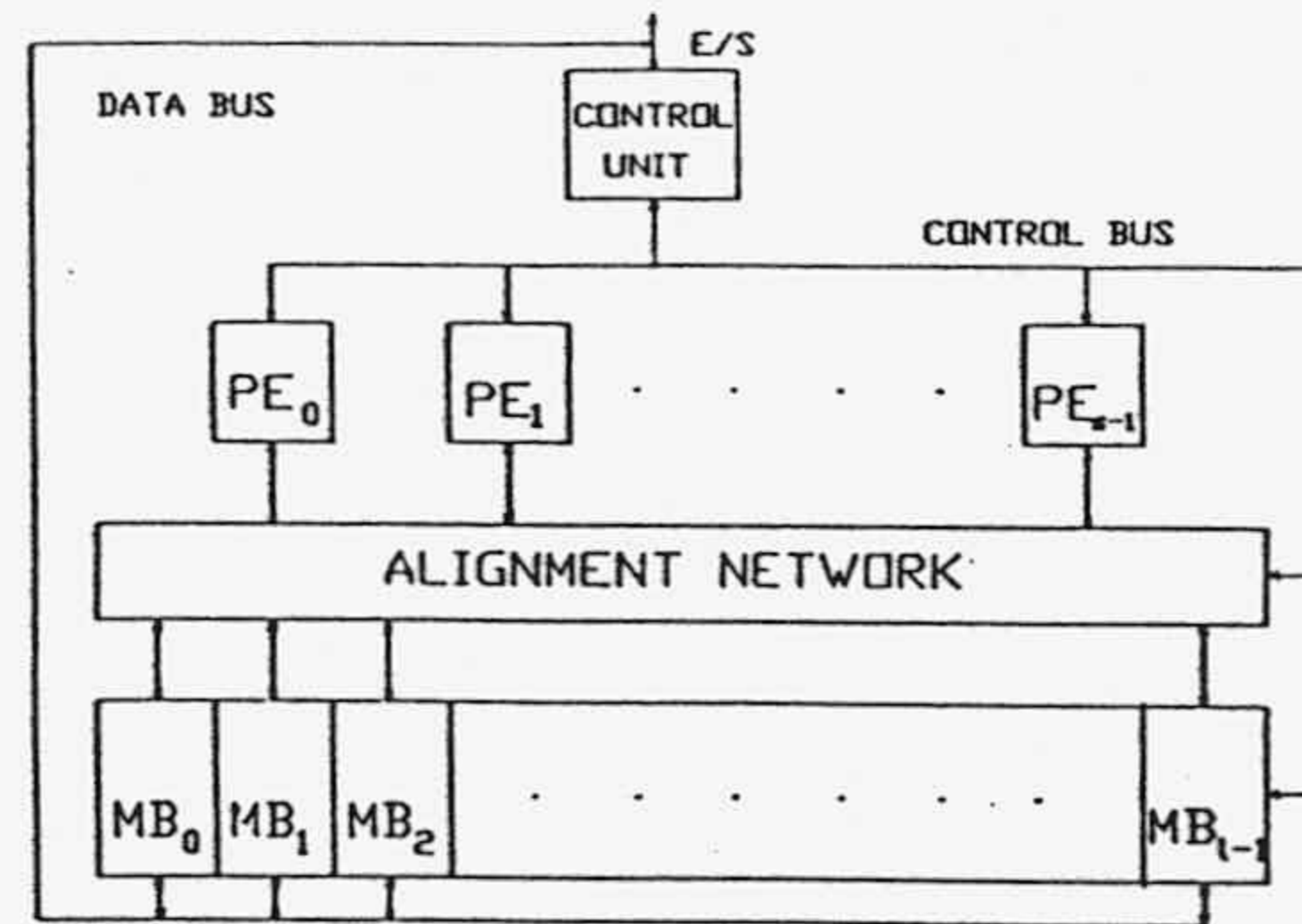


Figura 2.6. Una estructura SIMD.

Así, en el cálculo de la transformada de Tamari de una función $f \in F_p^m$ utilizando un SIMD, el número de referencias a memoria que realiza cada EP es $2pm$, por lo que crece linealmente con m , igual que el número de pasos del algoritmo. No obstante, el número de EPs y MBs necesarios para que eso sea así, crece exponencialmente. Si se dispone de menos de p^m MBs o de menos de p^{m-1} EPs, algunas matrices $T(p,1)$ deberían calcularse secuencialmente.

2.2.3. Complejidad de cálculo de la FTT.

Para la obtención de la complejidad de cómputo de la FTT sobre F_p^m se define como "operación elemental" una multiplicación seguida de una suma, ambas dentro del cuerpo de Galois de orden p .

Dada una función de F_p^m , la matriz de la transformada de Tamari, $T(p,m)$, verifica que:

$$T(p,m) = G(p,m) \dots G(p,m) = (G(p,m))^m \quad (2.22)$$

Teniendo en cuenta que la matriz $G(p,m)$ se obtiene a partir de $T(p,1)$ según (2.19), resulta evidente que de los p^m elementos que hay en cada una de las columnas de $G(p,m)$, sólo p son distintos de cero. Por consiguiente, la matriz $G(p,m)$ sólo tiene $p \cdot p^m$ elementos no nulos.

Si utilizamos (2.22), puesto que hay que realizar $m-1$ productos de la

matriz $G(p,m)$ por sí misma y cada producto implica p^{m+1} operaciones elementales, el número total de operaciones elementales es

$$N_R(p,m) = p^{m+1}(m-1). \quad (2.23)$$

En el caso de que se hubiera considerado que la matriz de la transformación $T(p,m)$ es una matriz de $p^m \times p^m$, sin tener en cuenta que hay elementos nulos, el número de operaciones elementales sería

$$N_C(p,m) = p^m \cdot p^m. \quad (2.24)$$

Así, definiendo la eficiencia, μ , del cálculo de la FTT frente al cálculo de la transformada de Tamari de manera similar a como se define en [KUN80] para la FFT y otras transformadas rápidas se obtiene

$$\mu = (N_C(p,m) - N_R(p,m)) / N_C(p,m) = 1 - (m-1)p^{-(m-1)} \quad (2.25)$$

observándose que, puesto que μ crece hacia 1 a medida que m aumenta, cuanto mayor es el número de entradas de la función más eficiente resulta utilizar la FFT para obtener su espectro de Tamari.

Según lo expuesto, la complejidad de cálculo de la Transformada de Tamari que se ha definido mediante su algoritmo rápido es similar a la que poseen otras transformaciones ampliamente utilizadas, como la de Fourier y la de Walsh [KUN80]. La FTT constituye un procedimiento rápido de cálculo que facilita la obtención de todos los coeficientes del espectro de Tamari. No obstante, en ciertas situaciones puede interesar calcular sólo algunos de los coeficientes espectrales. Si la dependencia de cada coeficiente respecto de los valores de la función permitiera elaborar un procedimiento de cálculo lo suficientemente rápido, la reducción del costo computacional podría ser considerable. En la siguiente sección se estudia la obtención independiente de los coeficientes del espectro de Tamari.

2.3. Obtención aislada de los coeficientes espectrales de Tamari.

En este apartado se relacionan los coeficientes del espectro de Tamari con los valores de la función de los que dependen. Esto nos permite determinar el número de combinaciones de entrada para las que es necesario conocer la

respuesta del circuito para calcular cada coeficiente.

Una función $f \in F_p^1$ de una variable $y_1 \in C_p$ se puede representar mediante un vector de p componentes

$$[f_1(y_1)] = [f_1(0) \ f_1(1) \ \dots \ f_1(p-1)] \quad (2.26)$$

donde la componente $f_1(j)$ del vector es igual al valor de la función cuando la variable $y_1 = j$.

A partir de (2.26), una función $f \in F_p^m$ de m variables, y_1, y_2, \dots, y_m se puede representar mediante el vector

$$[f(y_1, y_2, \dots, y_m)] = [f_1(y_1)] \otimes f_2(y_2) \otimes \dots \otimes [f_m(y_m)] \quad (2.27)$$

en el que el valor de la función en el punto $(b_1, b_2, \dots, b_m) \in C_p^m$ se expresa como

$$f(b_1, b_2, \dots, b_m) = f_1(b_1) \cdot f_2(b_2) \cdot \dots \cdot f_{m-1}(b_m) \quad (2.28)$$

Según esto, como el espectro de Tamari de $f \in C_p^m$ se obtiene a partir de

$$[f^*] = [f] \cdot T(p, m) = [f] \cdot (T(p, 1) \otimes \dots \otimes T(p, 1))$$

considerando (2.24) se tiene

$$[f^*] = ([f_1] \otimes \dots \otimes [f_m]) \cdot (T(p, 1) \otimes \dots \otimes T(p, 1))$$

y si se aplican las propiedades del producto de Kronecker [HUR85]:

$$[f^*] = ([f_1] \cdot T(p, 1)) \otimes \dots \otimes ([f_m] \cdot T(p, 1)) \quad (2.29)$$

Es decir, el coeficiente $f^*(\beta_1, \dots, \beta_m)$ del espectro de Tamari se puede calcular mediante la expresión:

$$f^*(\beta_1, \dots, \beta_m) = ([f_1] \cdot [T(p, 1)_{\beta_1}]^T) \otimes \dots \otimes ([f_m] \cdot [T(p, 1)_{\beta_m}]^T) \quad (2.30)$$

donde con $[T(p, 1)_j]^T$ se ha notado la columna j -ésima de la matriz $T(p, 1)$. Según se deduce de la ecuación (2.30), la dependencia respecto de todas las variables está fijada únicamente por la matriz base de la transformación, $T(p, 1)$. Como ejemplo, considerese una función de F_3^3 de la que se desea calcular el coeficiente $f^*(2, 0, 1)$ del espectro de Tamari. Para ello, se parte de la matriz $T(3, 1)$:

$$[f_1(0) \ f_1(1) \ f_1(2)] \cdot \begin{bmatrix} 1 & 0 & 2 \\ 0 & 2 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

A partir de aquí, utilizando (2.28) y (2.30) se obtiene la expresión del coeficiente de Tamari correspondiente.

$$\begin{aligned}
 f^*(2,0,1) &= [f_1(0) \ f_1(1) \ f_1(2)] \cdot [2 \ 2 \ 1]^T \otimes \\
 &\quad \otimes [f_2(0) \ f_2(1) \ f_2(2)] \cdot [1 \ 0 \ 0]^T \otimes \\
 &\quad \otimes [f_3(0) \ f_3(1) \ f_3(2)] \cdot [0 \ 2 \ 2]^T = \\
 &= (2 \cdot f_1(0) \otimes 2 \cdot f_1(1) \otimes f_1(2)) \cdot f_2(0) \cdot (2 \cdot f_3(1) \otimes 2 \cdot f_3(2)) = \\
 &= f(001) \otimes f(002) \otimes f(101) \otimes f(102) \otimes 2 \cdot f(201) \otimes 2 \cdot f(202)
 \end{aligned}$$

Además, el número de valores de la función de los que depende un coeficiente espectral dado se conoce de forma inmediata a partir de la matriz $T(p,1)$: basta contar el número de elementos distintos de cero que hay en cada una de sus columnas. Por ejemplo, en el caso de la matriz $T(3,1)$ que nos ocupa, si el índice del coeficiente espectral es igual a 0 habrá que considerar una entrada en la variable correspondiente, si es 1 hay que considerar dos entradas para esa variable y si es 2 habrá una dependencia respecto a tres entradas. Finalmente habrá que multiplicar los índices de dependencia en cada variable. Así, para $f^*(2,0,1)$ habrá que considerar $3 \cdot 1 \cdot 2 = 6$, como efectivamente se obtuvo.

Se tiene, por tanto, una forma sencilla de determinar la dependencia de un coeficiente respecto de los valores de la función, y el número de valores de los que depende.

Una vez definida la Transformada de Tamari, presentada la FTT, y analizadas la complejidad de cálculo que implica y la dependencia de cada coeficiente respecto de los valores de la función, en la sección 2.4 se estudia su utilización en el test de circuitos digitales.

2.4.El Test de Circuitos utilizando la FTT.

Supóngase un circuito lógico que presente un defecto. Esto significa que la función que sintetiza el circuito es diferente de la que sintetizaría el mismo circuito sin defectos. Como se indicó al comienzo de este capítulo, los procedimientos espectrales de test se basan en una transformación que asocia unívocamente un espectro a cada función. Así, puesto que un defecto cambia la función, también modificará sus coeficientes espectrales. Con este planteamiento, el objetivo es relacionar los cambios en los coeficientes del espectro con los defectos que los producen, definiendo una signatura para el circuito.



En lo que respecta a la aplicación de la transformada de Tamari al test de circuitos multivaluados, el punto a) se ha considerado en la sección 2.2, al realizar el análisis de complejidad de la transformada. Se ha mostrado que al disponerse de un algoritmo rápido de cálculo, esa complejidad no es mayor que para otras herramientas espectrales anteriormente utilizadas. En cuanto al punto b), en la sección 2.3 se ha indicado la forma de calcular el número de entradas de las que depende un coeficiente espectral dado. El punto c) se aborda en este apartado.

Como ejemplo de las técnicas espectrales de test de circuitos, se puede utilizar el test del circuito de la figura 2.7. En la Tabla 2.1 se muestran los valores de un subconjunto de coeficientes espectrales de Tamari sobre F_2^4 para la función sin faltas (columna f) y para la función afectada por cada una de las faltas de anclaje simples no redundantes (columnas α/i). Estos coeficientes definen la signatura de la función a testear frente a faltas de anclaje simples, ya que para cada una de ellas cambia al menos un coeficiente espectral del conjunto. Además, los valores de la función que se necesitan para calcular todos los coeficientes son $f(0)$, $f(1)$, $f(2)$, $f(3)$, $f(4)$, $f(5)$, $f(6)$, $f(7)$, $f(8)$ y $f(9)$, por lo que se ha determinado también el conjunto de patrones de test, formado por 9 del total de 16 entradas posibles. En la figura 2.8 se ilustra una forma sencilla, aunque costosa en hardware, de implementar la circuitería de test.

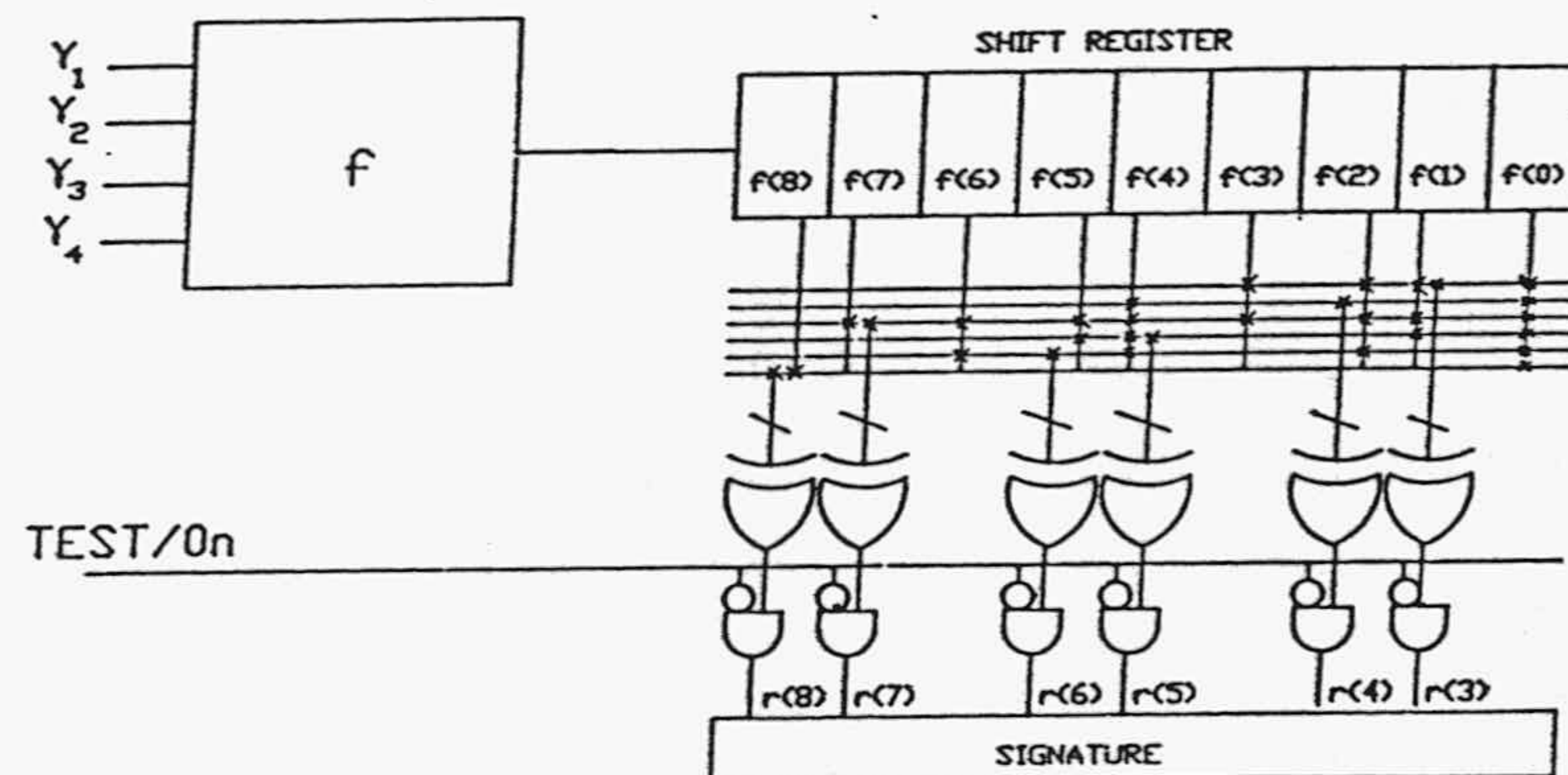


Figura 2.8. Circuito añadido para el test del circuito de la figura 2.7 con una signatura de coeficientes espectrales de Tamari.

A continuación se pasa a estudiar los aspectos que presenta el test espectral de circuitos digitales mediante la FTT. En la sección 2.4.1 se demuestran una serie de teoremas que permiten conocer la forma en que un defecto de anclaje afecta al espectro de Tamari y en la 2.4.2 se describe un

procedimiento para la determinación de la signatura que permita la detección de las faltas de anclaje simples.

2.4.1. Transformación de la FTT con las faltas de anclaje.

En este apartado se presenta una serie de teoremas que permiten establecer el conjunto de coeficientes del espectro de Tamari de una función que se ven afectados por los defectos de anclaje simples y múltiples en las entradas o en los nudos internos del circuito.

Sea un circuito combinacional que realiza la función $f(Y) \in F_p^m$. Utilizando (2.8), esta función se puede expresar en términos de las funciones de Tamari como:

$$\begin{aligned} f(Y) &= \sum_{w=0}^N r_w \cdot (y_1^{w_1} \cdot y_2^{w_2} \cdot \dots \cdot y_m^{w_m}) \\ \text{con } W &= (w_1, w_2, \dots, w_m) \in C_p^m; \\ Y &= (y_1, y_2, \dots, y_m) \in C_p^m; \\ w_i &\in C_p; N = p^m - 1 \end{aligned} \quad (2.31)$$

Teorema 2.2: Una falta de anclaje a $\alpha \in C_p$ en la entrada y_1 del circuito hace que se anulen todos los coeficientes del espectro de Tamari, r_w , con índice, $W = (w_1, \dots, w_1, \dots, w_m)$, tal que $w_1 \neq 0$.

Demostración: Teniendo en cuenta (2.31), si la variable de entrada y_1 está anclada al valor $\alpha \in C_p$, los términos producto de la forma $y_1^{w_1} \cdot \dots \cdot y_1^{w_1} \cdot \dots \cdot y_m^{w_m}$, con $w_1 \neq 0$ pasan a ser los términos $y_1^{w_1} \cdot \dots \cdot \alpha^{w_1} \cdot \dots \cdot y_m^{w_m}$.

Así, en la expresión de la función afectada por la falta de anclaje no aparecerán términos producto del tipo $y_1^{w_1} \cdot \dots \cdot y_1^{w_1} \cdot \dots \cdot y_m^{w_m}$ con $w_1 \neq 0$, y, por consiguiente, son nulos los coeficientes espectrales que tengan índice W con $w_1 \neq 0$. ■

El teorema 2.2 establece un criterio para seleccionar el conjunto de coeficientes espectrales que detectan las faltas de anclaje en las entradas. Si la entrada anclada es la y_1 , cualquier coeficiente espectral distinto de cero cuyo índice W contenga un valor de w_1 distinto de cero puede servir para detectar la falta. Así, en un circuito con m entradas hace falta un máximo de m coeficientes espectrales, uno por entrada, que verifiquen el teorema 2.1 para detectar todas las faltas de anclaje en las entradas. Un corolario inmediato del teorema 2.2 es el siguiente:

Corolario 2.1: Si en el espectro de Tamari de una función $f \in F_p^m$ existe un coeficiente $r_w \neq 0$ tal que su índice $W=(w_1, w_2, \dots, w_m)$ verifica que $w_i \neq 0$ para todo $i=1, \dots, m$, ese coeficiente espectral es suficiente para detectar cualquier falta de anclaje simple o múltiple en las entradas. ■

Una vez consideradas las faltas de anclaje en las entradas, pasamos a considerar las faltas en los nudos internos al circuito. Para éstas, se puede aplicar el siguiente teorema:

Teorema 2.3: Dada una línea g en un circuito combinacional que implementa la función $f(Y) \in F_p^m$, un fallo de anclaje a $\alpha \in C_p$ en la línea g es detectable mediante el coeficiente del espectro de Tamari $f^*(\beta)$, con $\beta=(\beta_1, \dots, \beta_m)$, si y sólo si $f^*(\beta) \neq \sum_{\sigma=0}^{p-1} \alpha^\sigma * h^*(\beta, \sigma)$, donde $h^*(\beta, \sigma)$ denota al coeficiente espectral de $h(Y, g(Y))$ con índices $(\beta_1, \dots, \beta_m, \sigma)$.

Demostración: Dada una línea interna del circuito, el valor de esa línea es una función de las entradas Y , $g(Y)$, donde $g(Y) \in F_p^m$ y describe el comportamiento del nudo interno de circuito considerado.

Se define una función de $m+1$ variables $h \in F_p^{m+1}$ que, para las entradas de la forma $(Y, g(Y))$, verifique que $f(Y) = h(Y, g(Y)) \forall Y \in C_p^m$. Entonces

$$[f(Y)] = [h(Y, g(Y))] = [G_0(g)] \otimes [h(Y, 0)] \# \dots \# [G_{p-1}(g)] \otimes [h(Y, p-1)] \quad (2.32)$$

$$\forall Y \in C_p^m, g \in C_p$$

con $G_j(g) = 1$ si $g=j$,
 $= 0$ en otro caso. $G_j \in F_p^1, g \in C_p, (j=0, \dots, p-1)$

Cuando se produce un error de anclaje a α en g , g/α , el circuito pasa a sintetizar la función $f_{g, \alpha}(Y)$. Si se considera la expresión (2.32), queda

$$[f_{g, \alpha}(Y)] = [G_0] \otimes [h(Y, \alpha)] \# \dots \# [G_{p-1}] \otimes [h(Y, \alpha)] = [h(Y, \alpha)]$$

El espectro de Tamari de $h(Y, g)$ está compuesto por p^{m+1} coeficientes. Sus elementos se pueden notar como $h^*(\beta, \sigma)$, con $\sigma=0, 1, \dots, (p-1)$ igual a la potencia con que aparezca la variable $n+1$ correspondiente a g en la función base, y $\beta=(\beta_1, \dots, \beta_n)$ es la n -tupla con componentes iguales a las potencias con que aparezcan en la función de Tamari correspondiente las n variables de Y .

Cuando la entrada g de $h(Y, g)$ está anclada a α , se tiene que todos los coeficientes espectrales $h^*(\beta, \sigma)$ con $\sigma \neq 0$ se anulan y los que tienen $\sigma=0$ se

transforman en los $h_{g,\alpha}^*(\beta,0)$, según la expresión:

$$h_{g,\alpha}^*(\beta,0) = \sum'_{\sigma=0}^{p-1} \alpha^\sigma \cdot h^*(\beta,\sigma) \quad (2.33)$$

Por otra parte, como $f(Y)=h(Y,g(Y))$, el espectro de $f_{g,\alpha}(Y)$ cuando la línea interna g está anclada a α , $[f_{g,\alpha}^*]$, es igual al espectro de $h(Y,\alpha)$. Así:

$$f_{g,\alpha}^*(\beta) = \sum'_{\sigma=0}^{p-1} \alpha^\sigma \cdot h^*(\beta,\sigma) \quad (2.34)$$

y los coeficientes espectrales de $[f^*]$ que permiten la detección de la falta de anclaje de la línea g a α deben verificar

$$f^*(\beta) \neq \sum'_{\sigma=0}^{p-1} \alpha^\sigma \cdot h^*(\beta,\sigma) \quad \blacksquare$$

Como continuación del teorema 2.3, en el caso de que g sea función de un conjunto de variables tal que $f(Y)=h(g(Y1),Y2)$, siendo $Y1 \cup Y2=Y$ e $Y1 \cap Y2=\emptyset$, $Y1=(y_1, \dots, y_r)$, $Y2=(y_{r+1}, \dots, y_m)$, se puede aplicar el siguiente teorema:

Teorema 2.4: Un error de anclaje a α en la línea g se detecta si y sólo si para algún conjunto de índices $\beta=(\beta_1, \dots, \beta_r)$ correspondientes a las variables de $Y1$ con al menos uno de sus elementos, β_i , distinto de cero se verifica que $f^*(\beta,\sigma) \neq 0$, para cualquier $\sigma=(\sigma_{r+1}, \dots, \sigma_m)$.

Demostración: En el espectro de f debe haber algún coeficiente no nulo $f^*(\beta,\sigma)$ con índice β tal que $\beta_i \neq 0$. Si no fuera así, eso significaría que f no depende de ninguna de las variables de g y, por consiguiente, una falta de anclaje en g no afectaría al comportamiento de f .

Cuando se produce una falta de anclaje en g , la función errónea que sintetiza el circuito no depende de las variables de $Y1$, luego los coeficientes $r(\beta,\sigma)$ con β conteniendo algún índice distinto de cero se harán nulos. ■

Otra forma alternativa de obtener información acerca de los cambios en los coeficientes espectrales se basa en la simulación del circuito para cada una de las faltas simples y el cálculo de los coeficientes espectrales para la función que sintetiza el circuito defectuoso [RUI89]. Esta posibilidad es fácil de programar y conduce a una información exhaustiva de los coeficientes que cambian, pero el tiempo que se precisaría la hace cada vez más impracticable a medida que la complejidad del circuito aumenta.

2.4.2. Coeficientes espectrales para el test de faltas de anclaje simples.

Utilizando los teoremas 2.2, 2.3 y 2.4 se puede elaborar un procedimiento para obtener un conjunto de patrones y la correspondiente signatura de test del circuito frente a los defectos que se modelan como faltas de anclaje simples.

Supóngase un circuito digital combinacional con r nudos (si un nudo actúa sobre s puertas se cuenta s veces) que sintetiza una función de m variables p valuadas, $f \in F_p^m$. Para determinar si el circuito está libre de faltas de anclaje simples en las entradas o en los nudos internos del circuito se podrían seguir, en principio, estos pasos:

- a) Para las faltas de anclaje en las entradas del circuito:
 - a.1) Determinar los p^m coeficientes del espectro de Tamari de la función. Entre los coeficientes no nulos, excepto $f^*(0)$, se pueden seleccionar los que detectan las faltas de anclaje simple en las entradas.
 - a.2) Para cada coeficiente espectral, $f^*(\beta)$, se define el conjunto $R(\beta)$ de las faltas de anclaje que puede detectar. Actualizar los conjuntos R para todos los coeficientes espectrales distintos de cero utilizando el teorema 2.2.
- b) Para las faltas de anclaje en los nudos internos del circuito:
 - b.1) Obtener los p^m coeficientes del espectro de Tamari para cada uno de los nudos del circuito según la expresión (2.34).
 - b.2) Para cada una de las faltas de anclaje en nudos internos, mediante los teoremas 2.3 ó 2.4, según el circuito, se determinan los coeficientes de Tamari que sufren cambios y se actualizan los $R(\beta)$ correspondientes.
- c) Determinar el conjunto óptimo de coeficientes espectrales que detectan todas las faltas de anclaje. Para ello se utilizan los p^m conjuntos $R(\beta)$, $\beta=0,1,\dots,p^m-1$. Los coeficientes espectrales seleccionados constituyen la signatura del circuito, $S(f)$.

Para completar a.1) y a.2) se necesita una memoria de $2mp^m \log_2(p)$ bits puesto que hay $2m$ posibles faltas de anclaje en las entradas y hay que considerar p^m coeficientes espectrales que pueden tomar valores comprendidos entre 0 y $p-1$. El número de pasos de cálculo que se precisan es mp^{m-1} para calcular la transformada de Tamari de la función y p^m-1 comparaciones con 0

(una por coeficiente) aplicando el teorema 2.1 para determinar qué faltas de anclaje en las entradas puede detectar. En cuanto a los pasos b.1) y b.2), se necesitarán $2(r-m)p^m \log_2(p)$ bits, como máximo, para almacenar los conjuntos $R(\beta)$, ya que hay $2r-2m$ faltas de anclaje simples en los nudos internos; un total de $2(r-m)(m+1)p^{m+1}$ pasos de cálculo para determinar los coeficientes espectrales según (2.34) en todas las faltas simples; y $2(r-m)p^m$ comparaciones para determinar los $R(\beta)$. En resumen, la complejidad de cálculo de a) y b) es proporcional a $2rmp^{m+1}$ y es necesario almacenar $2rp^{m+1} \log_2(p)$ bits.

Queda por obtener la complejidad del paso c). Para ello hay que precisar lo que se entiende por conjunto óptimo de coeficientes para la signatura, teniendo en cuenta los dos aspectos siguientes:

- i) El número de entradas distintas que hay que aplicar para determinar el valor de todos los elementos de la signatura. Este es, precisamente, el número de patrones de test necesarios, $NT(S(f))$.
- ii) El número de coeficientes que constituyen el síndrome, $N(S(f))$.

El conjunto de coeficientes óptimo es aquél que minimiza $NT(S(f))$ y $N(S(f))$. Sin embargo, según la forma en la que se vaya a implementar el test existirán ciertas restricciones en cuanto a:

- i) El número de coeficientes distintos que se pueden utilizar.
- ii) El número máximo de entradas a aplicar para obtener los coeficientes.

En el caso de que se pretenda utilizar un test BIST, interesa fundamentalmente disminuir el número de coeficientes, manteniendo el número de patrones a un nivel aceptable. En cambio, si se pueden ir comparando los coeficientes externamente, lo más importante es reducir el número de patrones. Si ninguna de estas posibilidades aporta soluciones aceptables, habría que pensar en otro procedimiento de test distinto al propuesto.

Suponiendo que se busca la signatura con menor número de coeficientes, el proceso a seguir en el paso c) sería:

- c.1) Ordenar los $R(\beta)$, con $\beta=0, \dots, 2^m-1$ y $R(\beta) \neq \emptyset$, según el número de faltas de anclaje que detectan: de mayor a menor número de faltas detectadas.
- c.2) Seleccionar el coeficiente que está en primer lugar de la lista. Si se han detectado todas las faltas, terminar, en caso contrario ir a c.3).
- c.3) Actualizar los $R(\beta)$, eliminando de ellos las faltas detectadas por el coeficiente $f^*(\sigma)$ seleccionado: $R(\beta) = R(\beta) - R(\sigma)$ ($\beta=0, \dots, 2^m-1$ $\beta \neq \sigma$). Ir a c.1).

La complejidad de este procedimiento es la siguiente:

- i) Para completar la etapa c.1) se necesitan p^m posiciones de memoria de

- $\log_2(p)$ bits para los coeficientes espectrales y m pasos de cálculo.
- ii) Para completar c.2) hacen falta p^{m+1} posiciones de memoria de $\log_2(p)$ bits y $p(r-m)(m+1)$ pasos de cálculo.
 - iii) Para completar c.3) se utilizan los conjuntos $R(\beta)$, $\beta=0,1,\dots,N$ ($N=p^m-1$), que se pueden representar por p^m palabras de rp bits y el número de pasos de cálculo que hacen falta es

$$\sum_{i=0}^{p^m-1} \{(p^m-i)\log_2(p^m-i)+(p^m-i-1)\} = (p^{2m}(\log_2(p^m)+1)-1)/2$$

que es aproximadamente igual a $p^{2m}\log_2(p^m)/2$.

Por consiguiente, para completar c.1) y c.2), se necesitan p^m+p^{m+1} posiciones de memoria de $\log_2(p)$ bits para almacenar los coeficientes, y $(p(r-m)(m+1)+m)$ pasos de cálculo. Para la etapa c.3) se necesitan p^m palabras de memoria de rp bits y un número de pasos igual a $p^{2m}\log_2(p^m)/2$.

Si el circuito a testear es multifuncional, hay que tener en cuenta que se pueden obtener en paralelo los coeficientes espectral de todas funciones. Por consiguiente, la complejidad que se añade en el caso multifuncional es la debida al cálculo de s conjuntos de coeficientes más (suponiendo que hay s funciones); es decir, c.1) y c.2) se realizarán en $s(p(r-m)(m+1)+m)$ pasos.

Evidentemente, debido a la complejidad exponencial respecto del número de entradas, a medida que éste aumenta, el tiempo y la memoria que se necesitarían para completar este procedimiento serían demasiado elevados. Por consiguiente, sólo es aplicable a circuitos moderadamente simples y con pocas entradas. Existen, no obstante, algunos circuitos en los que los teoremas 2.2, 2.3 y 2.4 permiten determinar de forma directa el conjunto de coeficientes a utilizar para constituir la signatura. Estos circuitos son los que se considerarán en las secciones siguientes.

2.5. Test de faltas de anclaje en circuitos producto/suma módulo-p.

Sea la función $f \in F_p^m$, cuya expresión en términos de la transformada de Tamari es $f(y_1, \dots, y_m) = \sum_{\beta} f^*(\beta_1, \dots, \beta_m) y_1^{\beta_1} \dots y_m^{\beta_m}$, y que está sintetizada según esa expresión mediante un circuito con niveles producto módulo-p/suma módulo-p. Por el teorema 2.2 se sabe que para detectar las faltas de anclaje en las entradas basta un conjunto de coeficientes espectrales tales que para toda variable y_i exista al menos un coeficiente para el que el índice β_i sea distinto de cero. En lo que a las faltas internas de anclaje a α ($\alpha=0, \dots, p-1$) se refiere, éstas corresponden a las salidas de las puertas producto módulo-p o suma módulo-p. Por consiguiente, una falta de anclaje en las mismas tendrá

el efecto de sustituir en la función sintetizada por el circuito defectuoso, uno o varios coeficientes espectrales por α .

Así pues, habrá que considerar todos y cada uno de los coeficientes, porque para cada uno de ellos se puede encontrar un nudo del circuito que sólo se detecta por un cambio en ese coeficiente: el nudo de salida de la puerta producto módulo- p correspondiente. Consecuentemente, en un circuito sintetizado según su desarrollo en términos de los coeficientes de Tamari, la signatura que permite detectar todas las faltas de anclaje (simples o múltiples) ha de estar constituida por todos los coeficientes distintos de cero. En la figura 2.9 se considera un ejemplo de este tipo de circuitos para el que se indica su signatura.

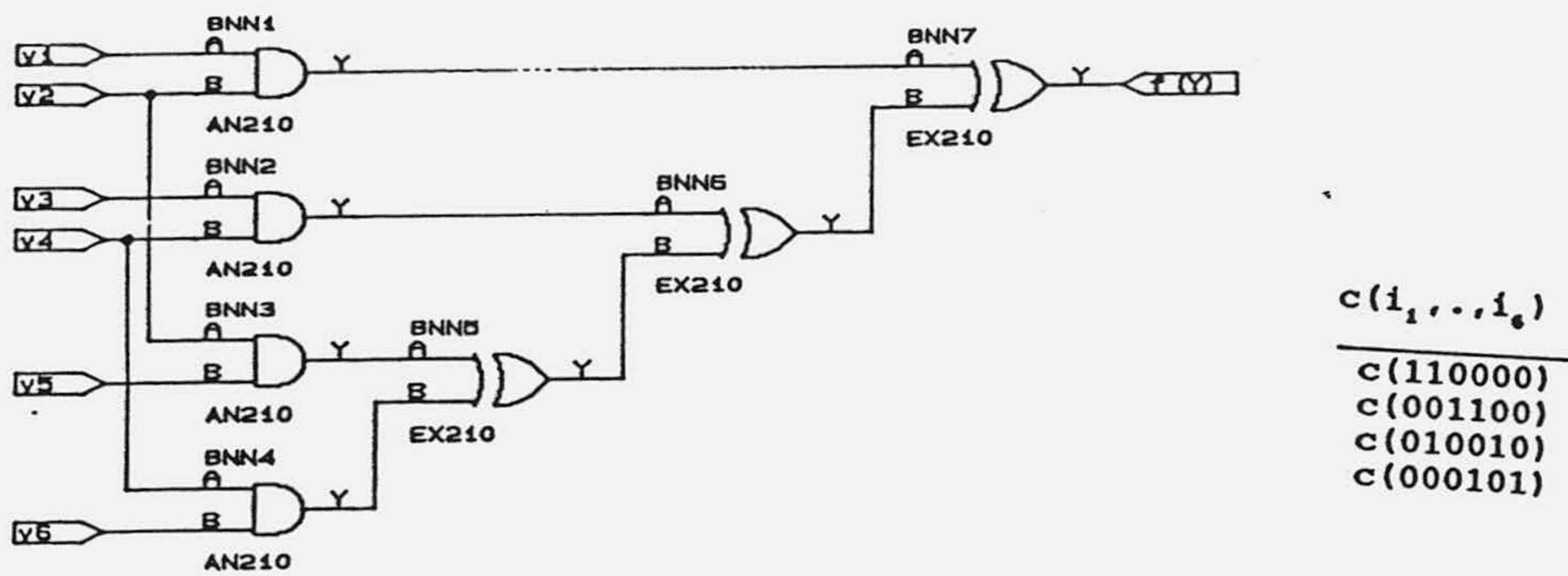


Figura 2.9. Ejemplo de circuito lógico.

2.6. Signatura espectral en redes de puertas suma módulo- p .

Si una función $f \in F_p^m$ se sintetiza mediante una red de puertas suma módulo- p , su expresión será de la forma:

$$f(y_1, \dots, y_{m-1}) = \alpha_1 \cdot y_1 \oplus \dots \oplus \alpha_m \cdot y_m \quad (\alpha_i \in C_p)$$

Una falta de anclaje en alguna de las entradas, y_j , hará que cambie el coeficiente $f^*(\beta_1, \dots, \beta_m)$, que verifique que $\beta_j = 1$ y $\beta_k = 0$ si $k \neq j$.

Las líneas interna de esta red son las salidas de una puerta suma módulo- p . Por consiguiente, una falta de anclaje en una de ellas haría desaparecer uno o varios de los sumandos de la función. Así, para detectar las faltas de anclaje simples y múltiples en una red de puertas suma módulo- p , la signatura ha de estar constituida por los m coeficientes espectrales $f^*(j)$ con $j = 1, p, p^2, \dots, p^{m-1}$. En la figura 2.10 se muestra un circuito de este tipo, junto con su signatura.

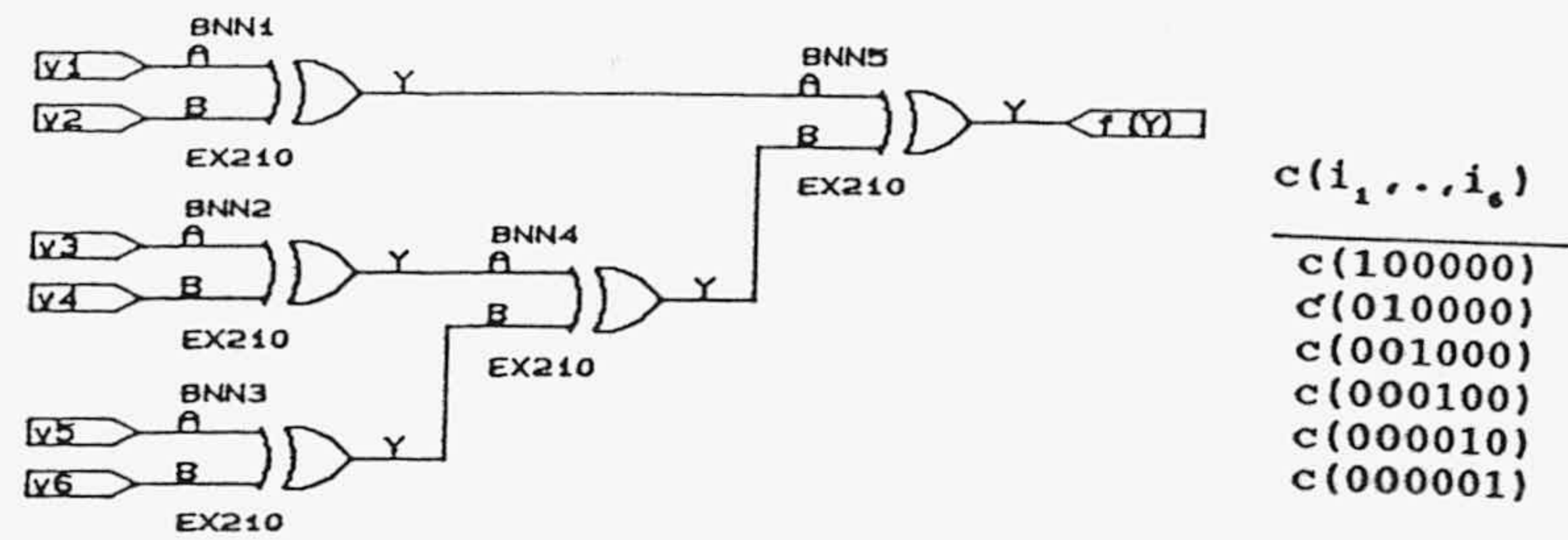


Figura 2.10. Ejemplo de circuito lógico de paridad.

2.7. Obtención de la signatura en redes sin reconvergencias.

Si la red que sintetiza la función $f \in F_p^m$ no posee ninguna reconvergencia, entonces para todos y cada uno de los nudos de la misma se puede aplicar el teorema 2.4. Así, si en el circuito existe algún nudo que depende de la variable y_j , para contruir la signatura hay que seleccionar el subconjunto de coeficientes espectrales de f que contenga al menos un coeficiente cuyo índice tenga la componente j -ésima, $\beta_j \neq 0$. De entre todos los subconjuntos de coeficientes que verifiquen esta condición, interesa elegir el óptimo, en los términos apuntados en 2.4.2. La figura 2.11 muestra un ejemplo de esta situación.

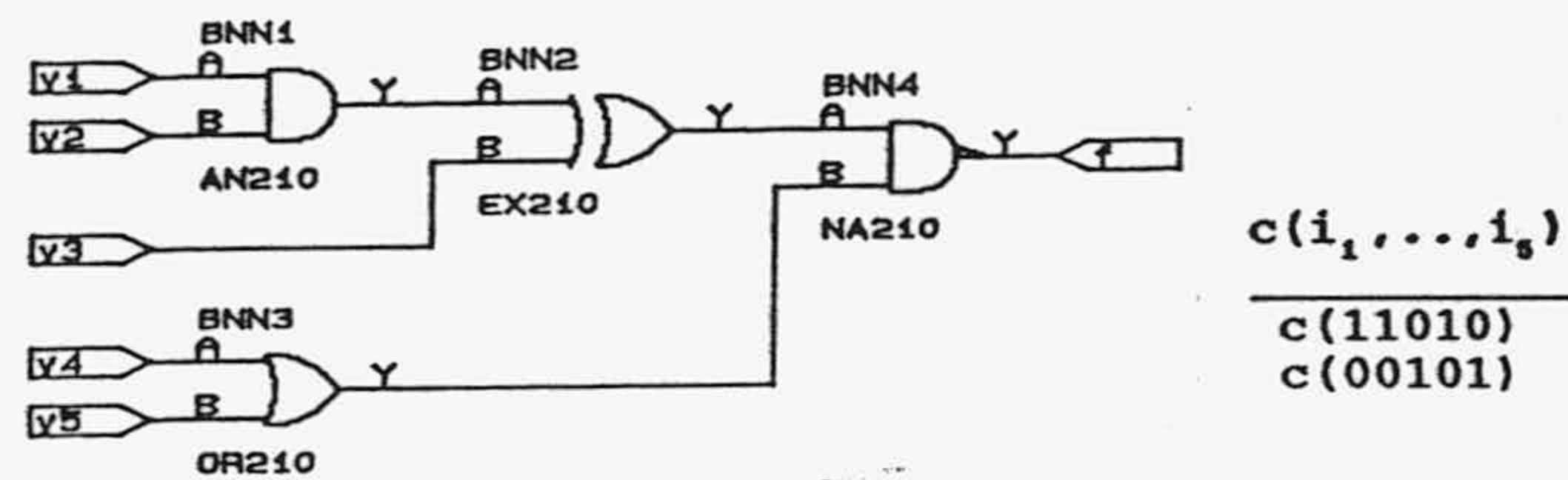


Figura 2.11. Ejemplo de circuito lógico sin reconvergencias.

2.8. Procedimiento espectral de generación de patrones de test.

Si se conoce la signatura espectral para la detección de un conjunto de faltas dado y como existe una forma de determinar los valores de la función de los que depende cada coeficiente, la misma signatura permite determinar el conjunto de patrones de test, tal como indica el siguiente teorema.

Teorema 2.5: Las combinaciones de entradas que se necesitan para obtener todos los coeficientes de la signatura son los patrones de test seleccionados.

Demostración: Basta tener en cuenta que si para alguna de las faltas detectadas por la signatura no se produjera ningún cambio en las salidas para ninguno de los patrones aplicados, tampoco se producirían para ella cambios en ninguno de coeficientes de la signatura, contrariamente a lo supuesto. ■

Naturalmente, el conjunto de patrones de test que se obtendría al aplicar el teorema 2.5, una vez conocida la signatura del circuito, no es el conjunto mínimo, pero permite determinar una cota superior del número de patrones necesario. Además, si se utiliza un simulador de faltas (como por ejemplo el incluido en el paquete software de desarrollo de ASICs LESSIM2) se puede visualizar qué patrones no aumentan el número de faltas detectadas a fin de suprimirlos.

2.9. Conclusion.

En este capítulo se ha definido la Transformada de Tamari. Se han demostrado sus propiedades mas significativas y se ha presentado un procedimiento rápido para su cálculo: la Transformada Rápida de Tamari (FTT).

La Transformada de Tamari proporciona de forma directa la síntesis de circuitos combinatoriales mediante puertas suma módulo-p y producto-módulo-p. En este capítulo se ha ilustrado la forma en que esto se puede llevar a cabo. Además, la Transformada de Tamari puede ser utilizada para el test de circuitos digitales comprobando los cambios en los coeficientes que las faltas ocasionan. En ese sentido, este capítulo presenta tres teoremas que establecen la relación entre los defectos modelados como faltas de anclaje simples y los cambios en los coeficientes espectrales de Tamari. Estos teoremas nos han permitido elaborar un procedimiento de determinación de patrones de test y de signatura que, aunque tiene una complejidad que crece exponencialmente con el número de entradas, crece linealmente con el número de nudos del circuito. Así, el procedimiento propuesto es superior a otros [GOE80, 81] en el caso de circuitos con pocas entradas pero con mucha complejidad interna.

También se ha apuntado la posibilidad de utilizar los procedimientos espectrales para determinar los conjuntos de patrones de test de los circuitos en procedimientos del tipo EPT/AEXH.

En el siguiente capítulo particularizaremos y ampliaremos los resultados al caso binario.

CAPITULO 3. LA TRANSFORMADA DE TAMARI EN CIRCUITOS BINARIOS.

En el presente capítulo se estudia la transformada de Tamari como una herramienta para el diseño y el test de los circuitos digitales binarios. Se trata, por consiguiente, de la Transformada de Tamari de las funciones de F_2^m . Los resultados del capítulo 2 se particularizan para el caso binario y se utilizan para determinar los cambios que los distintos tipos de faltas ocasionan en los coeficientes espectrales. En el capítulo 4 se elabora un procedimiento modular de cálculo de los coeficientes espectrales de Tamari que permite obtener los de un circuito a partir de los de los módulos que lo constituyen y que resulta de particular interés de cara a la utilización de procedimientos de test espectrales en circuitos complejos.

Recientemente han aparecido trabajos en los que se hace uso de los coeficientes del desarrollo de Reed-Muller de una función binaria para testear los circuitos digitales [DAR89a, DAR89b] y otros en los que se estudia la realización de funciones con sumas módulo-2 de productos, es decir con PLAs AND-EXOR, comparándose con la realización mediante PLAs AND-OR o considerando el problema de su minimización [GRE90, SAS90]. Esta situación parece denotar un cierto resurgimiento del interés en las redes de Reed-Muller o Redes AND-EXOR. Se había mostrado [RED72] que la generación de tests para las faltas de anclaje de líneas en redes AND-EXOR es sencilla y que el número de patrones de test que se necesitan para detectar todas las faltas de anclaje simples es menor que $3n+3$, donde n es el número de variables de entrada, o $2n+3$ para circuitos cuyas entradas se aplican a una sólo puerta lógica. Para detectar

faltas de anclaje simples y de cortocircuito (bridging), el número de patrones de test está acotado por $3n+5$ [DAR89b]. Si se utilizan los coeficientes de Reed-Muller en un procedimiento de test espectral hay que tener en cuenta que, frente a los de Walsh, la complejidad de cálculo es menor y sólo necesitan un bit para almacenarse. Además, se ha demostrado [DAR89a] que, a medida que el número de variables n aumenta, el número de coeficientes espectrales necesario para detectar todas las faltas de anclaje múltiples en las entradas tiende a $[0.75n]$ y la complejidad de cálculo tiende a $[1.25n]$ ($[x]$ parte entera de x).

En este capítulo se muestra que los coeficientes espectrales de Tamari sobre F_2^m coinciden con los coeficientes del desarrollo de Reed-Muller con polaridad cero de una función binaria. De esta manera, la transformada de Tamari se puede utilizar en la síntesis de funciones con PLAs AND-EXOR y en el test de circuitos que se basen en los coeficientes de Reed-Muller (ver Apéndice II).

3.1. La Transformada de Tamari de funciones binarias.

Los circuitos combinacionales digitales de m entradas sintetizan funciones que asocian elementos de C_2 a los elementos de C_2^m : se trata de las funciones de F_2^m . La base de Tamari para el espacio vectorial $V(F_2^m, C_2, \#, *)$ está constituida por las 2^m funciones:

$$T_i(Y) = y_1^{i_1} \cdot y_2^{i_2} \cdot \dots \cdot y_m^{i_m}, \quad i=0, \dots, r \quad (r=2^m-1) \quad (3.1)$$

$$\begin{aligned} \text{donde} \quad i &= i_1 + 2i_2 + 2^2i_3 + \dots + 2^{m-1}i_m, \quad i_j \in C_2 \\ Y &= y_1 + 2y_2 + 2^2y_3 + \dots + 2^{m-1}y_m, \quad y_j \in C_2 \\ y_j^0 &= 1, \quad y_j^1 = y_j \end{aligned}$$

Así pues, una función de F_2^m se puede expresar unívocamente como:

$$f(Y) = \sum'_{i=0}^r r_i \cdot (y_1^{i_1} \cdot y_2^{i_2} \cdot \dots \cdot y_m^{i_m}) \quad (3.2)$$

$$\text{con } Y = (y_1, y_2, \dots, y_m) \in C_2^m; \quad i=0, 1, \dots, r$$

siendo \sum' la suma módulo 2 sobre C_2 .

El desarrollo de $f \in F_2^m$ dado en (3.2) constituye la representación de Reed-Muller de polaridad cero de dicha función [DAR89a, GRE90]. Por tanto, el espectro de Tamari indica qué términos aparecen en la expansión de Reed-Muller de polaridad cero y la transformada de Tamari definida sobre F_2^m permite pasar de la representación como minterms a la representación en términos de los coeficientes de Reed-Muller de polaridad cero.

Aplicando los resultados del capítulo 2, la transformada de Tamari para

F_2^m se puede obtener a partir de m veces el producto de Kronecker de la matriz $T(2,1)$ por sí misma:

$$T(2,m) = T(2,1) \otimes \dots \otimes T(2,1) \quad (3.3)$$

donde

$$T(2,1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Como ejemplo, para las funciones binarias de tres variables, $f \in F_2^3$, la matriz de la transformación es:

$$T(2,3) = T(2,1) \otimes T(2,1) \otimes T(2,1) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

y las funciones de Tamari $T_2(Y)$, $T_5(Y)$ y $T_7(Y)$, por ejemplo, son:

$$T_2(Y) = Y_2 \quad T_5(Y) = Y_1 \cdot Y_3 \quad T_7(Y) = Y_1 \cdot Y_2 \cdot Y_3$$

Así, dada una función $f \in F_2^3$, por ejemplo $f(Y_1, Y_2, Y_3) = Y_1 + Y_2 Y_3$, para obtener su espectro de Tamari, se parte del vector que da la descripción de la función en términos de sus minterms, $[f(Y)] = [0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1]$, y multiplicándolo por $T(2,3)$ se obtiene el espectro de Tamari de la función:

$$[f(Y)] \cdot T(2,3) = [0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1]$$

y la expansión de polaridad cero de Reed-Muller será, por consiguiente:

$$f(Y) = Y_1 \oplus Y_2 \oplus Y_1 Y_2 \oplus Y_2 Y_3 \oplus Y_1 Y_2 Y_3$$

en donde se ha suprimido el punto del producto módulo-2 y \oplus denota la suma módulo-2 definidas sobre C_2 . En lo que sigue se mantendrá esta notación.

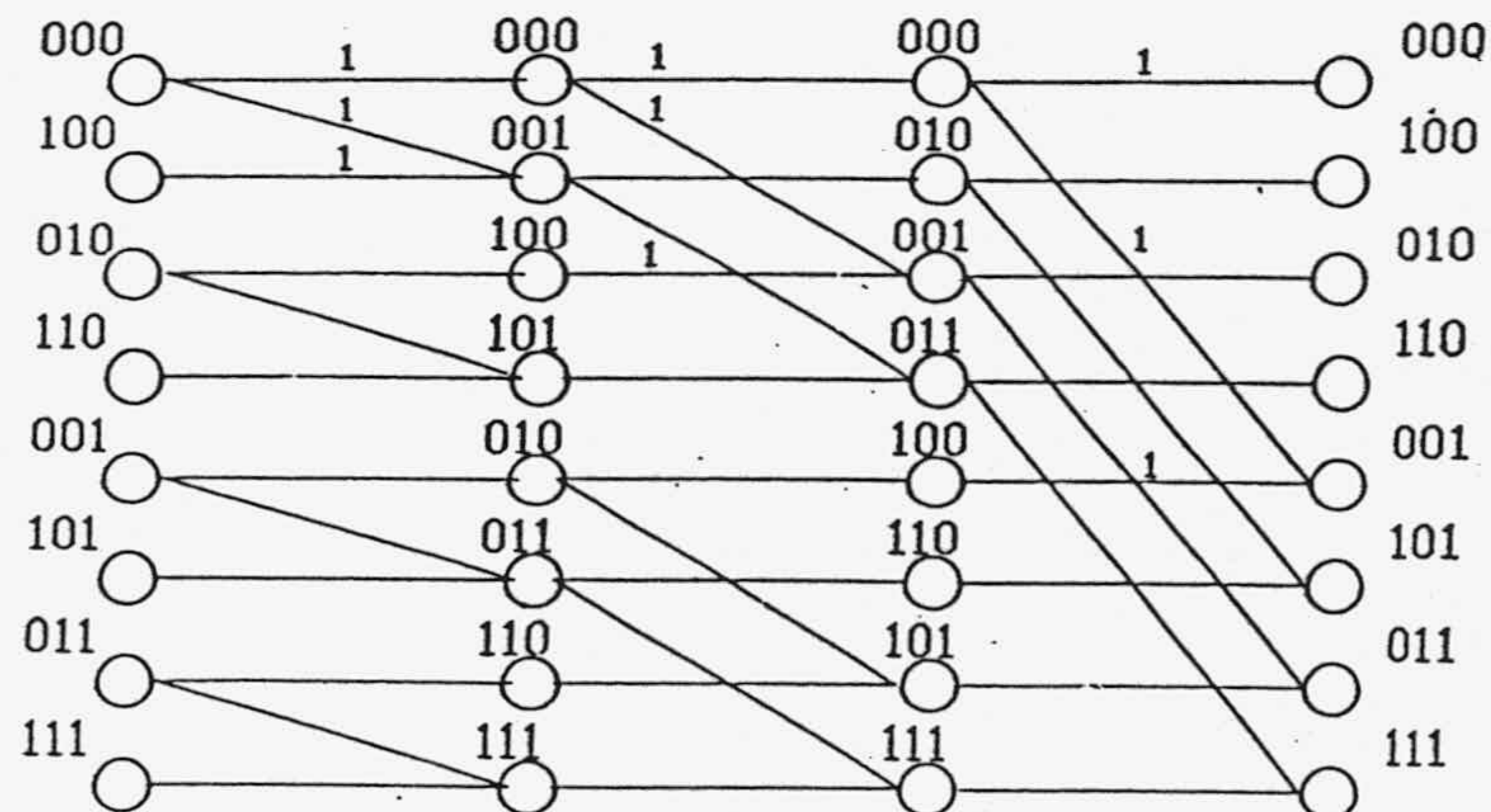


Figura 3.1. Grafo de Cálculo para la FTT de F_2^3 .

Por supuesto, el procedimiento de cálculo rápido para la transformada de Tamari (FTT) también se puede aplicar a $T(2,m)$. En la figura 3.1 se muestra el grafo de cálculo para la FTT de una función binaria de tres variables. El esquema general de la síntesis de una función mediante una red de Reed-Muller se puede ver en la figura 3.2.

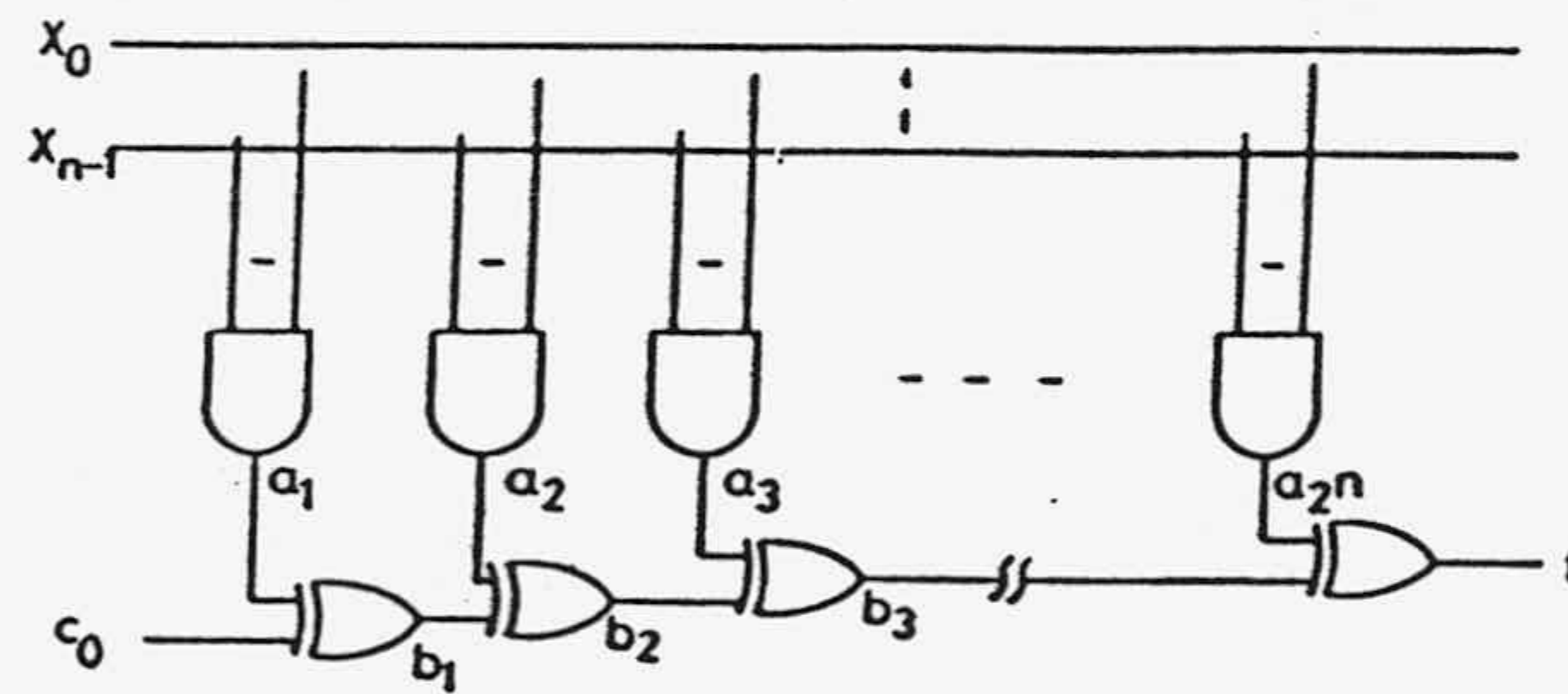


Figura 3.2. Circuito para la síntesis Reed-Muller.

3.1.1. Propiedades de los coeficientes espectrales.

El espectro de Tamari de las funciones de F_2^m posee una serie de propiedades que se presentan en esta sección. Algunas de esas propiedades son consecuencia directa de las propiedades generales de la transformada de Tamari presentadas en el capítulo 2; otras, en cambio, sólo son ciertas para F_2^m . Las propiedades que se presentan son útiles ya que facilitan el cálculo de los coeficientes espectrales al permitir identificar las combinaciones de entradas de las que dependen y relacionar los coeficientes espectrales conocidos de subcircuitos más simples del circuito que se considera (ver Apéndice III para la demostración de estas propiedades).

Dada una función $f \in F_2^m$ cuyo espectro está constituido por los elementos $\{r_f(i) / i=0,1,\dots,N\}$, $N=2^m-1$, se verifican las siguientes propiedades:

Propiedad 3.1: Para cualesquiera $i=(i_1, \dots, i_m)$ e $Y=(y_1, \dots, y_m)$ con $y_j \in C_2$ e $i_j \in C_2$ se verifica que

$$\sum_{j=0}^N T_j(i) T_j(Y) = \delta_{i,Y}$$

donde $\delta_{a,b}=1$ si $a=b$ y $\delta_{a,b}=0$ en otro caso. ■

Teniendo en cuenta que el elemento $t(i,j)$ de la matriz de la transformación sobre F_2^m verifica que $T_j(i)=t(i,j)$, esta propiedad establece que la matriz inversa de la transformación, T_2^m , es la misma matriz.

Propiedad 3.2: Para todo $i=(i_1, \dots, i_m)$ con $i_j \in C_2$ se verifica que

$$r_f(i) = \sum'_{Y=0}^N T_Y(i) f(Y) = \sum'_{Y \subseteq i} f(Y)$$

donde $Y=(y_1, \dots, y_m)$ con $y_j \in C_2$ e $Y \subseteq i$ si y sólo si $y_j \leq i_j \forall j=1, \dots, m$. ■

Esta propiedad permite identificar los valores de la función que intervienen en el cálculo de cada coeficiente espectral. Así pues, posibilita el cálculo de los coeficientes del espectro de Tamari de forma independiente, sin tener que calcular todo el espectro. Es consecuencia del resultado demostrado en la sección 2.3 para el caso general, $T(p,m)$. Como ejemplo, si se desea calcular el coeficiente espectral $r_f(0110)$ de una función $f(Y) \in F_2^4$ sólo hay que conocer el valor que toma la función para los valores $(0,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$ y $(0,1,1,0)$ de las variables (y_1, y_2, y_3, y_4) :

$$r_f(0,1,1,0) = f(0,0,0,0) \oplus f(0,1,0,0) \oplus f(0,0,1,0) \oplus f(0,1,1,0)$$

Propiedad 3.3: Si las funciones $f, g, h \in F_2^m$ verifican que $f(Y) = g(Y) \oplus h(Y)$, para todo $Y \in C_2^m$ se cumple que $r_f(i) = r_g(i) \oplus r_h(i)$, $\forall i=0,1, \dots, N$. ■

Esta propiedad no es otra que la de linealidad de la transformada de Tamari demostrada en la sección 2.1.

Propiedad 3.4: Si $f, g, h \in F_2^m$ verifican que $f(Y) = g(Y)h(Y)$, $\forall Y \in C_2^m$, entonces $r_f(i) = \sum'_{a \cup b = i} r_g(a)r_h(b)$, con $a=(a_1, a_2, \dots, a_m)$; $b=(b_1, b_2, \dots, b_m)$; $i=(i_1, i_2, \dots, i_m)$; $a_j, b_j, i_j \in C_2$ y definiéndose $a \cup b = (a_1 \cup b_1, \dots, a_m \cup b_m)$. ■

Facilita el cálculo de los coeficientes espectrales de una función producto de otras dos con espectros conocidos.

Propiedad 3.5: Si $f, g, h \in F_2^m$ verifican que $f(Y) = g(Y) \cup h(Y)$, $\forall Y \in C_2^m$, entonces

$$r_f(i) = r_g(i) \oplus r_h(i) \oplus (\sum'_{a \cup b = i} r_g(a)r_h(b))$$

Proporciona la misma información que la 3.4 pero en el caso de que la función sea la suma booleana de dos funciones.

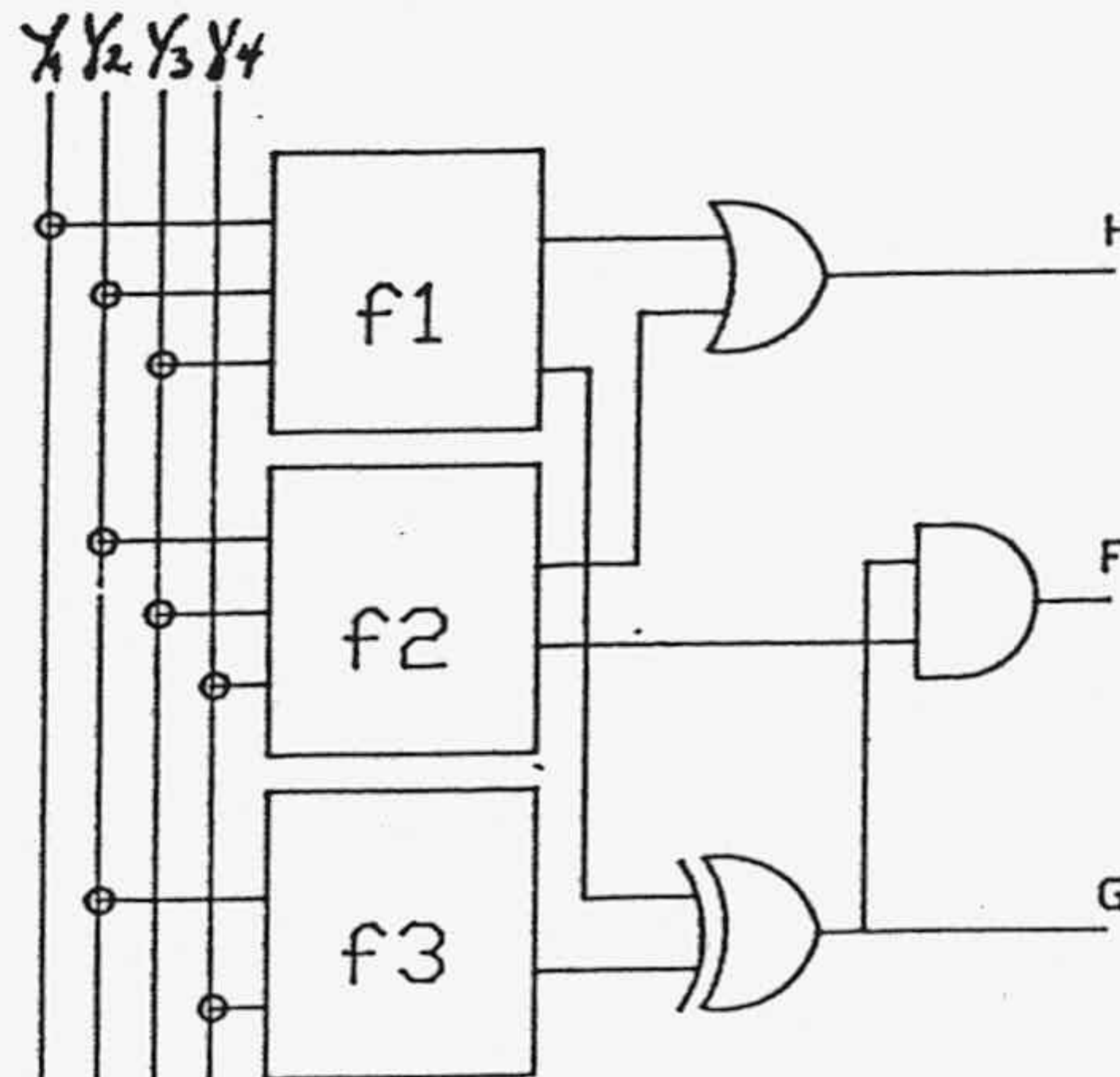


Figura 3.3. Circuito lógico de ejemplo.

Como ejemplo de aplicación de las propiedades 3.3, 3.4 y 3.5 se puede utilizar el circuito de la figura 3.3. En él los bloques f_1 , f_2 y f_3 corresponden a tres funciones cuyos coeficientes espectrales distintos de cero son:

$$\begin{aligned} & r_{f_1}(1000), r_{f_1}(0101), r_{f_1}(1010), r_{f_1}(1110) \\ & r_{f_2}(0010), r_{f_2}(0110), r_{f_2}(0111) \\ & r_{f_3}(0000), r_{f_3}(0101) \end{aligned}$$

y, utilizando la propiedad 3.3, los coeficientes distintos de cero de la función G son $r_G(0000)$, $r_G(1000)$, $r_G(1010)$, $r_G(1110)$; utilizando 3.4 los coeficientes distintos de cero en el espectro de F son $r_F(0010)$, $r_F(0110)$, $r_F(0111)$; y, finalmente, aplicando la propiedad 3.5 los coeficientes de la función H distintos de cero son $r_H(0010)$, $r_H(0101)$, $r_H(0110)$, $r_H(1000)$, $r_H(1010)$, $r_H(1110)$, $r_H(1111)$.

Propiedad 3.6: Si los espectros de $f, g, h \in F_2^m$ verifican que $r_f(i) = r_g(i)r_h(i)$, $\forall i=0,1,\dots,r$, se cumple que $f(Y) = \sum_{X \cup Z = Y} g(X)h(Z)$, con $X, Y, Z \in C_2^m$. ■

Esta propiedad es, en cierto modo, la inversa de la propiedad 3.4 ya que permite obtener la forma de operar dos funciones para obtener otra cuyo espectro sea el producto de los espectros de estas dos funciones. Como ejemplo, dadas las funciones $g, h \in F_2^3$ cuyos elementos espectrales distintos de cero son $r_g(011)$, $r_g(100)$, $r_g(110)$, $r_h(010)$, $r_h(011)$, $r_h(111)$, se pueden calcular los valores de la función f cuyo único coeficiente espectral distinto de cero es $r_f(011)$ mediante la propiedad 3.6. Concretamente

$$\begin{aligned} f(000) &= g(011)h(010) \oplus g(011)h(001) \oplus g(011)h(011) \oplus g(000)h(011) \oplus \\ & \oplus g(010)h(001) \oplus g(010)h(011) \oplus g(001)g(010) \oplus g(001)h(011) = 1 \end{aligned}$$

Una vez presentadas estas propiedades de los coeficientes espectrales de $T(2,m)$, en la siguiente sección se considera su utilización en el test de circuitos.

3.1.2. Obtención secuencial de los coeficientes espectrales.

En esta sección se describe la forma de calcular los coeficientes espectrales de la transformada de Tamari sobre F_2^m cuando no se dispone de todos los valores de la función simultáneamente. Se trata de ir generando coeficientes espectrales sin tener que esperar a recibir todos los valores de la función a transformar.

Dada $f(Y) \in F_2^m$, $Y = (y_1, \dots, y_m) \in C_2^m$, se supone que los valores de esa

función se obtienen ordenadamente, de forma que cuando se dispone de $f(i)$, $i=(i_1, \dots, i_m)$, se han recibido todos los valores $f(j)$, $j=(j_1, \dots, j_m)$ tales que

$$j=j_1 + \dots + j_m 2^{m-1} < i_1 + \dots + i_m 2^{m-1} = i,$$

y únicamente esos. Teniendo en cuenta la propiedad 3.2, los valores de la función que intervienen en el cálculo del coeficiente $r_f(\beta)$, $\beta=(\beta_1, \dots, \beta_m)$ son todos aquellos $f(j)$ tales que $j \subseteq \beta$, es decir $j \leq \beta$. Consiguientemente, el cálculo del coeficiente $r_f(\beta)$ se puede realizar una vez se tiene el valor de f correspondiente a la combinación $\beta=(\beta_1, \dots, \beta_m)$, siendo el procedimiento a seguir el que se detalla a continuación.

Algoritmo 3.1 (Transformada de Tamari secuencial)

1. Hacer $r_f(j)=0$ ($j=0, \dots, 2^m-1$).
2. Desde $i=0$ hasta $i=2^m-1$, $i=(i_1, \dots, i_m)$:
 - 2.1 Obtener $f(i)$,
 - 2.2 Para todo j ($j=i, \dots, 2^m-1$), tal que $i \subseteq j$ hacer $r_f(j)=r_f(j) \oplus f(i)$
 - 2.3 Se obtiene $r_f(i)$. ■

En este algoritmo, se precisan 2^m bits para almacenar la transformada, un bit por coeficiente. El número de veces que se accede a memoria, y por consiguiente la complejidad debida a la etapa 1.2 es

$$\sum_{i=0}^{2^m-1} 2^{m-1} \binom{m}{i} = 3^m$$

El paso 2.1 se refiere a la determinación del valor de la función a transformar. En el caso de que se pretenda obtener la transformada para la función que realiza un circuito determinado, este paso corresponde a la simulación funcional del circuito para la entrada $i=(i_1, \dots, i_m)$.

3.2. Los coeficientes espectrales en el test de circuitos binarios.

En esta sección se describen los cambios que distintos tipos de faltas habitualmente usadas para modelar los defectos de los circuitos lógicos pueden ocasionar en los coeficientes del espectro de Tamari sobre F_2^m . Se considerará que los circuitos se ven afectados únicamente por defectos que se modelan con faltas simples. Esta situación no se corresponde totalmente con la realidad [ORT89, WIL81, SET84, SHE85] pero se ha comprobado que un porcentaje muy elevado de los defectos modelados como faltas múltiples se detectan por los test generados considerando faltas simples [ABR80, BRE76, HUG86, JAC87, JHA86]. Esto justifica que sólo se consideren faltas simples. Como ya se vió

en el capítulo 2, la información acerca de los cambios que los defectos originan en los coeficientes espectrales se puede utilizar para:

a) Construir una signatura con el menor número de coeficientes espectrales afectados por las faltas a detectar.

b) Seleccionar un conjunto de combinaciones de entradas para formar un test que detecte esas faltas: aquellas entradas de las que dependan los coeficientes espectrales de la signatura.

En primer lugar, en la sección 3.2.1 se utilizan los resultados expuestos en la sección 2.4 para obtener criterios que permitan determinar qué coeficientes espectrales cambian debido a las faltas de anclaje simples en circuitos digitales binarios. A continuación, la sección 3.2.2 estudia el efecto de las faltas de cortocircuitado (**bridging**) en los coeficientes espectrales. Finalmente, en las secciones 3.2.3 a 3.2.6 se particularizan los resultados de las secciones 2.5 a 2.7 para circuitos binarios y se considera el test de PLAs AND-OR mediante los coeficientes de Tamari sobre F_2^m . Para las faltas de anclaje de transistor stuck-open en circuitos CMOS, ver Apéndice IV.

El cálculo de los coeficientes espectrales afectados por una falta determinada en circuitos complejos y con un número grande de entradas puede llevar un tiempo elevado y precisar una gran cantidad de memoria. En el capítulo 4, se describe una forma de relacionar los cambios en los coeficientes espectrales de los módulos de un circuito con los que sufren los coeficientes espectrales de la función que sintetiza el circuito.

3.2.1. Detección de las faltas de anclaje.

En esta sección se enuncia una serie de corolarios que permiten determinar los efectos de las faltas de anclaje simples sobre los coeficientes espectrales. Esos corolarios son consecuencia directa de los teoremas 2.2, 2.3 y 2.4 para el caso de la transformada sobre funciones binarias, por consiguiente no se detalla su demostración.

Dado el circuito combinacional que realiza la función $f(Y) \in F_2^m$ expresada en términos del espectro de Tamari $T(2,m)$ como

$$f(Y) = \sum_w r_w (y_1^{w_1} y_2^{w_2} \dots y_m^{w_m}) \quad (w=0, \dots, N) \quad (3.4)$$

con $w = (w_1, w_2, \dots, w_m)$, $w_i \in \{0, 1\}$,

se verifican los siguientes corolarios:

Corolario 3.1: Una falta de anclaje a 0 ó a 1 en la entrada y_1 del circuito hace que se anulen todos los coeficientes del espectro de Tamari, r_w , con índice $w=(w_0, w_1, \dots, w_{m-1})$ tal que exista algún $w_i=1$ ($i=0, \dots, m-1$). ■

Este corolario establece un criterio de selección de coeficientes espectrales para la detección de las faltas de anclaje en las entradas. Es la particularización del teorema 2.2 al caso binario.

Corolario 3.2: Si en el espectro de Tamari de una función $f \in F_2^m$, el coeficiente $r_N \neq 0$, entonces ese coeficiente es suficiente para detectar todas las faltas de anclaje simples o múltiples en las entradas al circuito. ■

Es consecuencia de la aplicación del corolario 2.1 al caso binario y proporciona un método para hacer un circuito testeable con un número mínimo de coeficientes frente a las faltas de anclaje en las entradas.

Corolario 3.3: Dada una línea interna, g , del circuito que realiza la función $f(Y)$, una falta de anclaje a α ($\alpha=0$ ó $\alpha=1$) en la línea interna g es detectable mediante el coeficiente del espectro de Tamari $r_f(\beta)$, con $\beta=(\beta_0, \beta_1, \dots, \beta_{m-1})$, si y sólo si $r_f(\beta) \neq r_h(\beta, 0)$ o $r_h(\beta, 1)$, con $r_h(\beta, 0)$ y $r_h(\beta, 1)$ denotando a los coeficientes espectrales de $h(Y, g(Y))$ con índices $(\beta_1, \dots, \beta_m, 0)$ y $(\beta_1, \dots, \beta_m, 1)$ respectivamente. ■

Permite la selección de coeficientes espectrales para detectar faltas de anclaje en los nudos internos de un circuito y se deriva del teorema 2.3.

Corolario 3.4: Si la línea interna g del circuito es tal que $f(Y)=h(g(Y1), Y2)$ siendo $Y=Y1UY2$ y $Y1 \cap Y2 = \emptyset$, entonces un error de anclaje a α en la línea g se detecta si y sólo si para algún conjunto de índices $\beta=(\beta_1, \dots, \beta_r)$ correspondientes a las variables de $Y1$ con algún $\beta_i=1$ se verifica que $r_f(\beta, \sigma) \neq 0$, para cualquier $\sigma=(\sigma_{r+1}, \dots, \sigma_m)$, $\sigma_i \in C_2$. ■

Este corolario constituye la particularización del teorema 2.4 y establece un criterio de selección de patrones para detectar las faltas de anclaje en nudos internos de circuitos que se puedan partir en subcircuitos con entradas disjuntas.

Como ejemplo de aplicación de estos corolarios se considera el circuito de la figura 3.4. La función que implementa dicho circuito es

$$f(y1, y2, y3, y4, y5) = y1y2 \oplus y1y3 \oplus y4y5 \oplus y1y2y4y5$$

y, por consiguiente, los coeficientes espectrales distintos de cero son $r_f(11000)$, $r_f(10100)$, $r_f(00011)$, $r_f(11011)$.

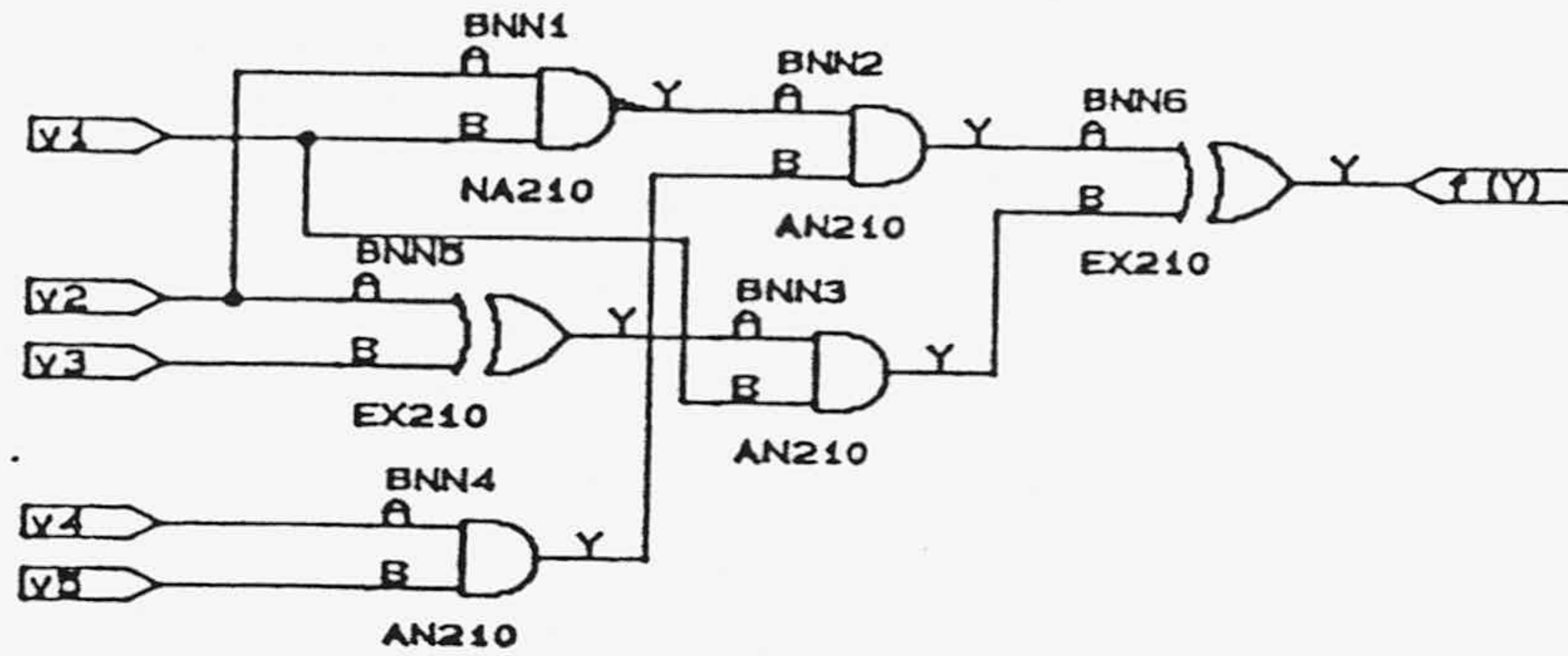


Figura 3.4. Ejemplo de circuito lógico.

Puesto que $r_f(11111)=0$, no se puede aplicar el corolario 3.2 y hará falta más de un coeficiente espectral para asegurar la detección de todas las faltas de anclaje en las entradas. Teniendo en cuenta el corolario 3.1, el coeficiente $r_f(11011)$ permite detectar todas las faltas de anclaje simples o múltiples en y_1, y_2, y_4 y y_5 , mientras que el coeficiente $r(10100)$ detecta las faltas de anclaje en y_3 . Para seleccionar un coeficiente espectral que detecte una falta de anclaje en la línea interna BNN3. Y se utilizará el corolario 3.3. La línea interna BNN3. Y realiza la función

$$g(y_1, y_2, y_3) = y_1 y_2 \oplus y_1 y_3$$

verificándose que

$$f(y_1, y_2, y_3, y_4, y_5) = h(y_1, y_2, y_3, y_4, y_5, g) = y_4 y_5 \oplus y_1 y_2 y_4 y_5 \oplus g$$

y según el corolario 3.3 se podrán utilizar los coeficientes espectrales $r_f(11000)$ y $r_f(10100)$ para detectar las faltas de anclaje en estas líneas, puesto que

$$r_f(11000) \neq f_h(110000) \oplus a r_h(110001)$$

$$r_f(10100) \neq f_h(101000) \oplus a r_h(101001)$$

Como ejemplo de aplicación del corolario 3.4 se consideran las faltas de anclaje de la línea BNN4. Y. En este caso la función que realiza esa línea depende de las variables y_4 e y_5 , que no actúan sobre ninguna otra puerta del circuito. Aplicando directamente el corolario 3.4, los coeficientes espectrales que permiten la detección de las faltas de anclaje en esa línea son $r_f(00011)$ y $r_f(11011)$.

3.2.2. Detección de las faltas de cortocircuito en las entradas.

Supóngase que un circuito realiza una función binaria de m variables, $f(Y)$, expresada en términos de los coeficientes espectrales de Tamari como:

$$f(Y) = \sum_{w=0}^N r_w (y_1^{w_1} y_2^{w_2} \dots y_m^{w_m}) \quad (3.5)$$

con $w = (w_1, w_2, \dots, w_m)$, $Y = (y_1, \dots, y_m)$
 $w_i, y_i \in \{0, 1\}$, $N = 2^m - 1$

Una falta de cortocircuito entre las entradas y_i e y_j se puede describir sustituyendo ambas variables por

$$y_i = y_j \quad \rightarrow \quad y_i y_j \quad (3.6)$$

si el circuito se ha realizado con lógica positiva, o por

$$y_i = y_j \quad \rightarrow \quad y_i + y_j = y_i \oplus y_j \oplus y_i y_j \quad (3.7)$$

si se ha realizado con lógica negativa.

Si hay que utilizar (3.6), los coeficientes espectrales de la función que implementa el circuito afectado por la falta (r'_w) en términos de los de la función sin faltas (r_w) serán

$$\begin{aligned} r'_{w(i,j)} &= r_{w(i)} \oplus r_{w(j)} \oplus r_{w(i,j)} \\ r'_{w(i)} &= 0 \\ r'_{w(j)} &= 0 \end{aligned} \quad (3.8)$$

donde

$$\begin{aligned} w(i) &= (w_1, \dots, w_i = 1, \dots, w_j = 0, \dots, w_m) \\ w(j) &= (w_1, \dots, w_i = 0, \dots, w_j = 1, \dots, w_m) \\ w(i,j) &= (w_1, \dots, w_i = 1, \dots, w_j = 1, \dots, w_m) \end{aligned}$$

Por consiguiente, para detectar una falta de cortocircuitado que se manifiesta según (3.6) se puede utilizar $r_{w(i)}$, si es igual a 1; $r_{w(j)}$, si es igual a 1; y $r_{w(i,j)}$, si $r_{w(i)} \oplus r_{w(j)}$ es 1.

Si se debe utilizar (3.7), la relación entre coeficientes espectrales del circuito sin defectos y con defectos es

$$\begin{aligned} r'_{w(i,j)} &= r_{w(i)} \oplus r_{w(j)} \oplus r_{w(i,j)} \\ r'_{w(i)} &= r_{w(i)} \oplus r_{w(j)} \oplus r_{w(i,j)} \\ r'_{w(j)} &= r_{w(i)} \oplus r_{w(j)} \oplus r_{w(i,j)} \end{aligned} \quad (3.9)$$

donde $w(i)$, $w(j)$, $w(i,j)$ como en (3.8).

Así, se puede utilizar $r_{w(i)}$, si $r_{w(j)} \oplus r_{w(i,j)}$ es 1; $r_{w(j)}$, si $r_{w(i)} \oplus r_{w(i,j)}$ es 1; y $r_{w(i,j)}$, si $r_{w(j)} \oplus r_{w(i)}$ es 1.

La demostración de (3.8) y (3.9) se obtiene de forma directa al sustituir (3.6) y (3.7), respectivamente, en (3.5) y comparar la nueva expresión de (3.5) con la que se tenía antes.

Considérese como ejemplo que en el circuito de la figura 3.4 se produce un cortocircuito entre las entradas y_1 e y_2 . Si el circuito se sintetiza con

lógica positiva, la falta se detectaría comprobando si el coeficiente espectral $r_f(10100)$ se anula. En el caso de que se emplee lógica negativa el coeficiente $r_f(10011)$ permitirá la detección, ya que

$$r_f'(10011) = r_f(10011) \oplus r_f(01011) \oplus r_f(11011) = 1 \neq r_f(10011)$$

3.2.3. Detección de faltas en circuitos AND-EXOR.

Sea la función $f \in F_2^m$ sintetizada según su expresión en términos de la transformada de Tamari sobre F_2^m , o lo que es igual, en términos de los coeficientes de Reed-Muller:

$$f(Y) = \sum_{w=0}^N r_w (Y_1^{w_1} Y_2^{w_2} \dots Y_m^{w_m})$$

con $w = (w_1, w_2, \dots, w_m)$, $Y = (Y_1, \dots, Y_m)$
 $w_i, Y_i \in \{0, 1\}$

Teniendo en cuenta lo establecido para F_p^m en la sección 2.5, para asegurar que se detectan las faltas de anclaje en todos los nudos hay que considerar todos los coeficientes espectrales distintos de cero que aparezcan en la expresión de la función. No obstante, si sólo se pretende detectar las faltas de anclaje simple o múltiple en las entradas, sólo hace falta seleccionar de entre esos coeficientes un conjunto tal que para toda variable y_i exista un coeficiente espectral con índice w_i distinto de cero, tal y como se establece en las secciones 2.2 y 2.5.

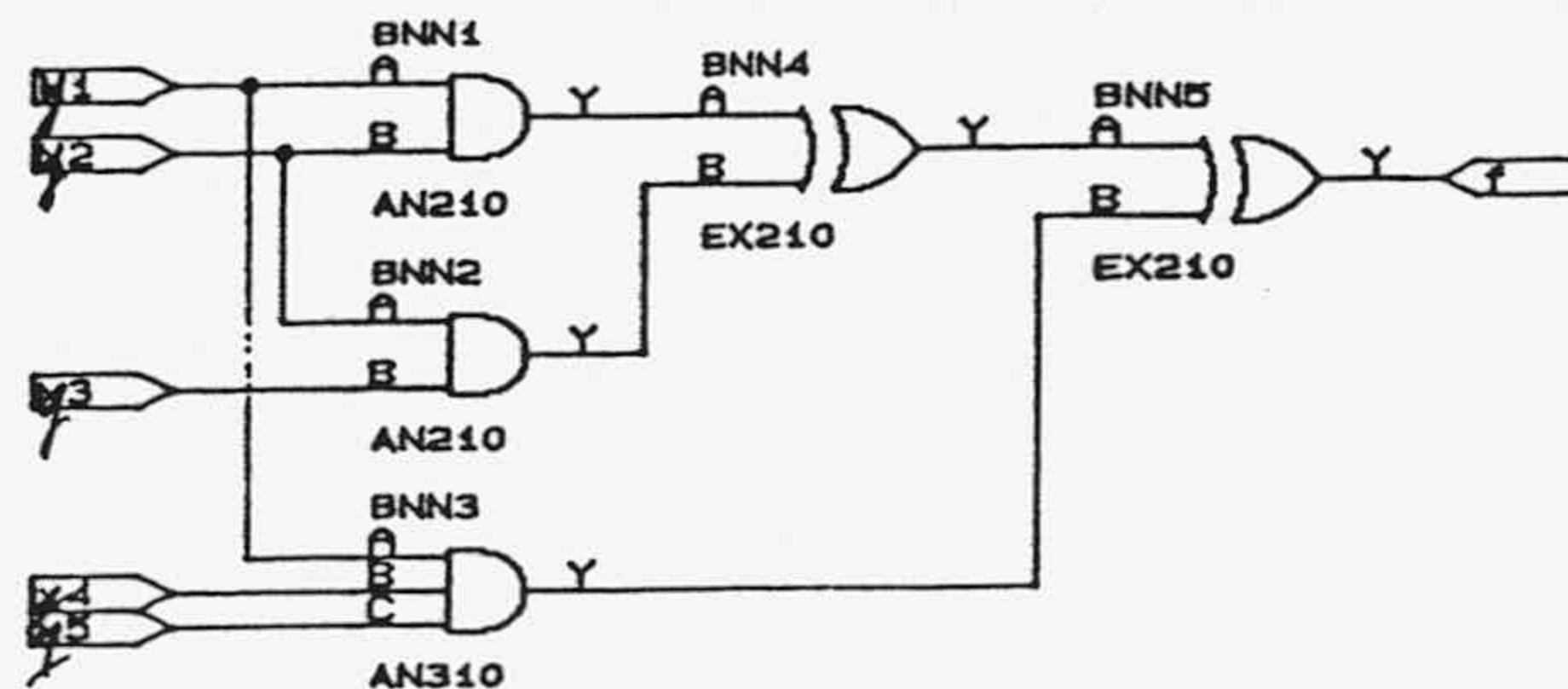


Figura 3.5. Ejemplo de circuito AND EXOR.

Como ejemplo, se considera el circuito de la figura 3.5, que sintetiza la función $f(y_1, y_2, y_3, y_4, y_5) = y_1 y_2 \oplus y_2 y_3 \oplus y_1 y_4 y_5$. El número máximo de patrones necesarios para detectar todas las faltas de anclaje simples es $3n+3$ [DAR89b], es decir 18. Ahora bien, teniendo en cuenta que las faltas de anclaje simples

o múltiples modificarán uno o varios de los coeficientes espectrales $r(11000)$, $r(01100)$ y $r(10011)$, los patrones necesarios para detectar cualquier defecto de anclaje simple o múltiple se podrán encontrar buscando las entradas de las que dependen esos coeficientes. Eso indica que como máximo se necesitarán 12 patrones de error para detectar todas las faltas simples o múltiples:

(00000), (10000), (01000), (11000), (00100), (01100)

(00010), (00001), (10010), (00011), (10001), (10011)

por lo que la cota superior del número máximo de patrones es menor que $3n+3$ en este caso. Si se pretenden detectar las faltas de anclaje a partir de los cambios en los coeficientes espectrales, hay que comprobar los tres coeficientes espectrales.

3.2.4. Detección de faltas en circuitos de paridad.

Dada una función de paridad de m variables $f(y_1, \dots, y_m) = y_1 \oplus \dots \oplus y_m$, cualquier falta de anclaje simple o múltiple y de cortocircuitado en el circuito que la implementa se detecta comprobando si ha habido cambios en alguno de los m coeficientes espectrales r_w con $w=1, 2, 4, \dots, 2^{m-1}$.

La demostración se obtiene directamente de la sección 2.6 mediante su particularización al caso binario. Teniendo en cuenta las combinaciones de entradas que se necesitan para determinar estos coeficientes, a partir de la propiedad 3.2, el número de patrones es $m+1$. Es decir se necesitarían los patrones de entrada $(0, 0, \dots, 0)$, $(0, \dots, y_i, \dots, 0)$, $i=1, \dots, m$.

Como ejemplo, en el circuito de la figura 3.6 todas las faltas de anclaje se pueden detectar con los siete patrones (000000), (100000), (010000), (001000), (000100), (000010), (000001), de los que dependen los seis coeficientes espectrales $r(10000)$, $r(01000)$, $r(00100)$, $r(00010)$ y $r(00001)$.

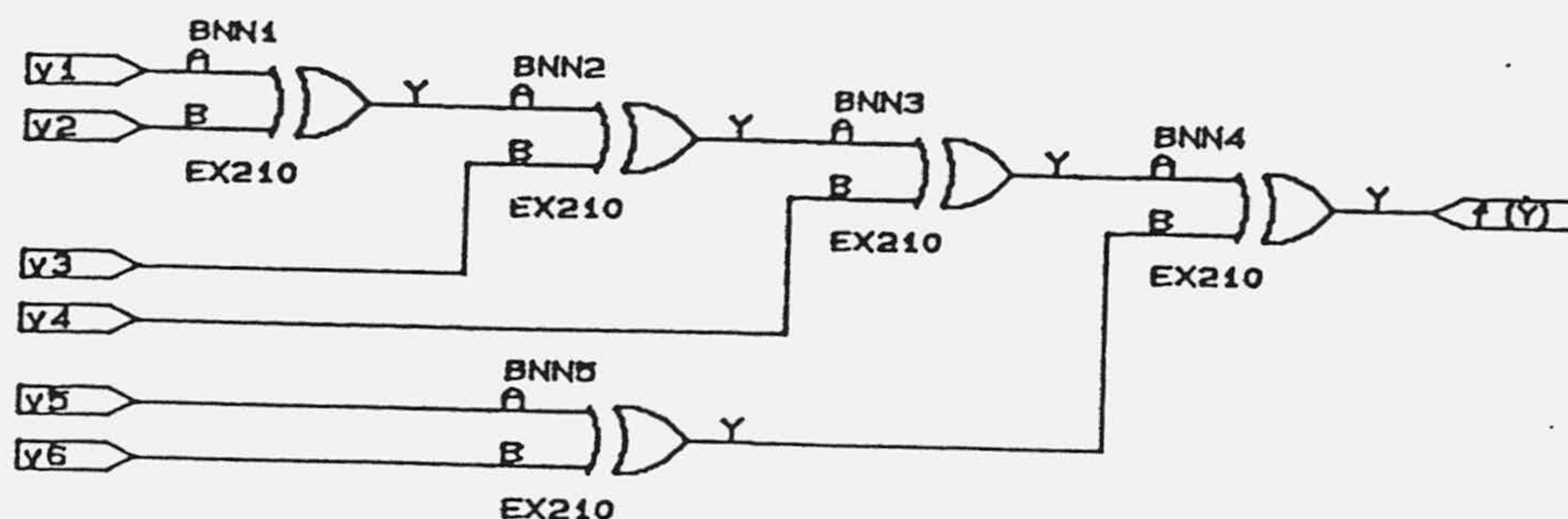


Figura 3.6. Ejemplo de circuito de paridad.

3.2.5. Detección de faltas en circuitos sin reconvergencias.

Si el circuito que sintetiza la función $f(Y) \in F_2^m$ no tiene ninguna reconvergencia, se puede aplicar el teorema 2.4 a todos y cada uno de sus nudos, tal y como se indicó en la sección 2.7. Teniendo esto en cuenta, para detectar una falta de anclaje en un nudo g se elige entre aquellos coeficientes espectrales distintos de cero con algún índice $w_i \neq 0$ suponiendo que la variable y_i es una de las que depende el valor del nudo g . Una vez determinados los coeficientes que pueden detectar cada falta, el problema consiste en seleccionar de entre ellos el número mínimo de coeficientes que aseguren una cobertura suficiente de faltas de anclaje simple.

En el ejemplo de la figura 3.7, el circuito no presenta ninguna línea con reconvergencias y sintetiza la función de ocho variables:

$$f(y_1, \dots, y_8) = y_1 y_3 \oplus y_2 y_3 \oplus y_4 \oplus y_1 y_3 y_4 \oplus y_2 y_3 y_4 \oplus y_5 y_6 \oplus y_7 \oplus y_5 y_6 y_7 \oplus y_8$$

Los coeficientes espectrales $r(01100000)$, $r(10110000)$, $r(00001110)$ y $r(00000001)$ constituyen un conjunto necesario y suficiente para la detección de todas las faltas de anclaje del circuito. Puesto que estos coeficientes dependen de 18 combinaciones de entrada, este es el número máximo de patrones de test necesarios para la detección de todas las faltas de anclaje.

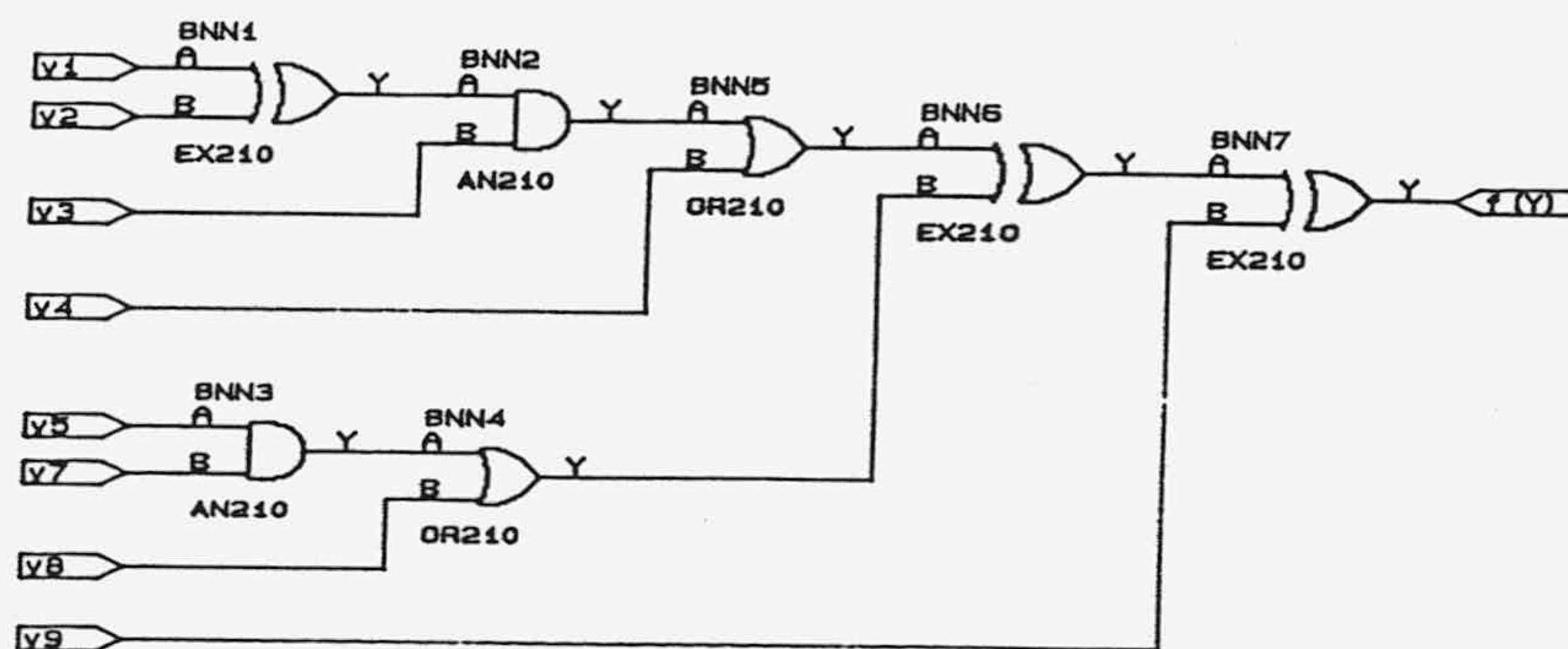


Figura 3.7. Ejemplo de circuito sin reconvergencias.

3.2.6. Detección de faltas en PLAs AND-OR.

En esta sección se supone que todos los términos producto que realiza la PLA son mutuamente disjuntos. Esta suposición aparece con bastante frecuencia en la literatura sobre el test de PLAs [FUJ81, SON80, WAN79, DAR89a]. La consecuencia de esta suposición es que la PLA que se considera no será la PLA de mínimo tamaño para realizar un conjunto de funciones dado. Hay

que tomar la suposición como una condición que debe cumplirse para que la PLA sea fácilmente testable. Así pues, las N funciones $f_1(Y), \dots, f_N(Y) \in F_2^m$ de m variables $Y=(y_0, y_1, \dots, y_{m-1})$, $y_i \in C_2$ se podrán expresar como

$$f_i(Y) = P_0 \oplus \dots \oplus P_j \oplus \dots \oplus P_{L_i} \quad (3.9)$$

donde los P_j son los productos de variables, complementadas y sin complementar, que intervienen en la línea de producto j -ésima y se pueden expresar de la forma

$$P_j = y_{\sigma_1} \dots y_{\sigma_r} y'_{\sigma_{r+1}} \dots y'_{\sigma_s} = y_{\sigma_1} \dots y_{\sigma_r} (1 \oplus y_{\sigma_{r+1}}) \dots (1 \oplus y_{\sigma_s}) \quad (3.10)$$

donde $\sigma_1 \dots \sigma_s$ es un subconjunto de índices de variables en el que cada índice puede aparecer una vez a lo sumo. A partir de (3.9) y (3.10) se deduce que los coeficientes espectrales de la función a los que contribuye el producto P_j son todos aquellos cuyos índices $w=(w_0, \dots, w_{m-1})$ cumplen que

$$\begin{aligned} w_k &= 0 \text{ si } k \notin \sigma_t \quad (t=1, \dots, s) \\ w_k &= 1 \text{ si } k = \sigma_t \quad (t=1, \dots, r) \end{aligned} \quad (3.11)$$

es decir, contribuye a 2^{s-r} coeficientes espectrales.

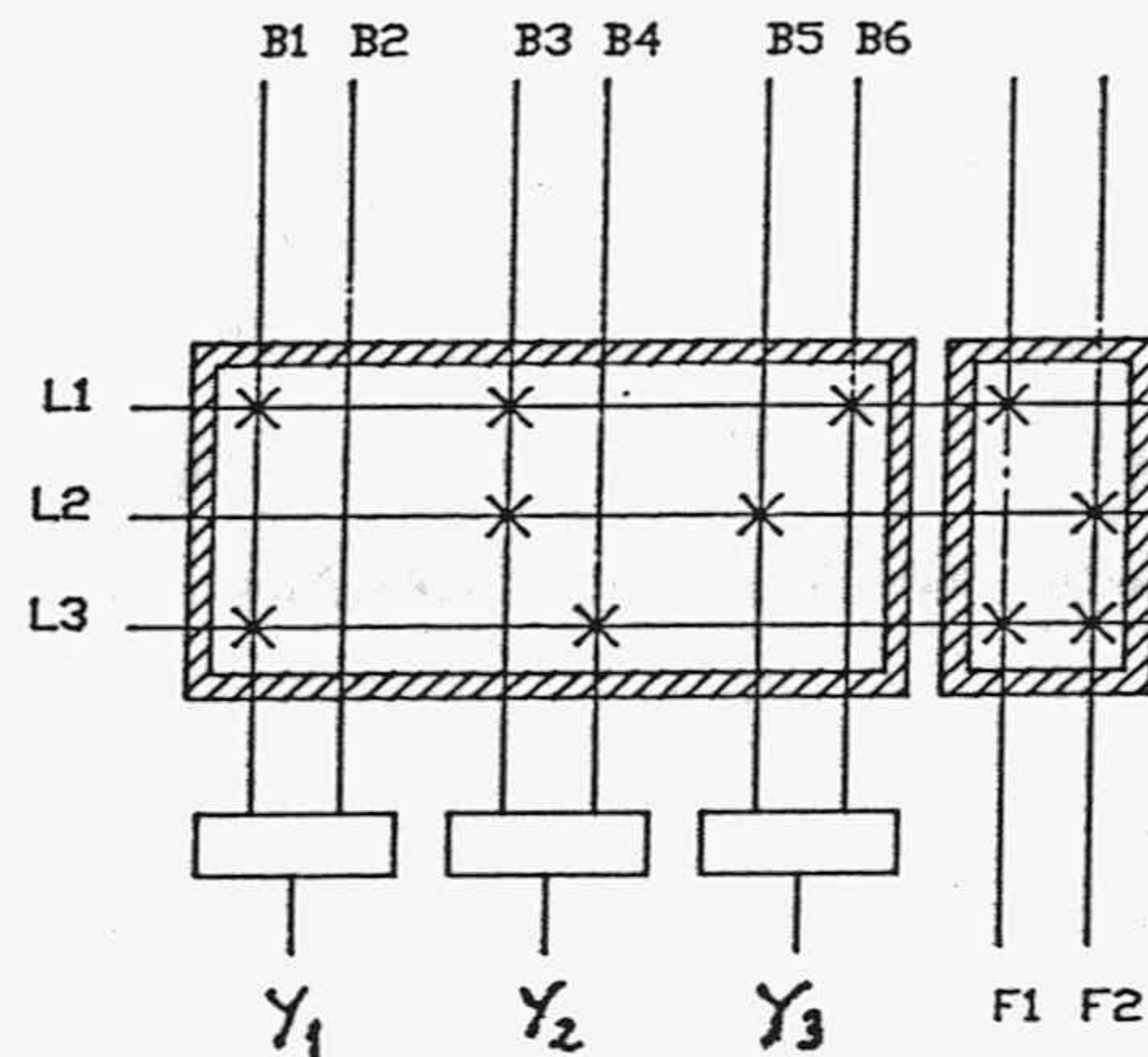


Figura 3.8. Ejemplo de PLA AND-OR.

Como ejemplo se utilizará la PLA AND-OR de la figura 3.8. Esta PLA realiza dos funciones cuyos términos producto son mutuamente disjuntos:

$$F_1(y_1, y_2, y_3) = y_1 y_2' + y_1 y_2 y_3' = y_1 y_2' \oplus y_1 y_2 y_3' = y_1 \oplus y_1 y_2 y_3$$

$$F_2(y_1, y_2, y_3) = y_1 y_2' + y_2 y_3 = y_1 \oplus y_1 y_2 \oplus y_2 y_3$$

A continuación se estudian los cambios que las faltas de la PLA producirán en los coeficientes espectrales de Tamari. Las faltas que se

consideran son las de anclaje en las líneas de producto, salida y bit, y las de pérdida de contacto o contacto extra en los panos AND y OR [LIU88, ORT90, TRE85]. Estas faltas permiten modelar los defectos que se producen en las PLAs, excepto aquellos que en PLAs CMOS corresponden a transistores continuamente conduciendo (falta de stuck-on) y dan lugar a salidas indeterminadas desde el punto de vista del nivel lógico, y las faltas de transistor continuamente abierto en los transistores de pull-up que originan un comportamiento secuencial de la PLA y necesitan un par de patrones de test para detectarse [LIU88, ORT90].

(a) Falta de anclaje en la línea de salida i -ésima: Si la línea i -ésima está anclada a α , se tiene que $f_i(Y)=\alpha$, y por consiguiente, todos los coeficientes espectrales se hacen cero excepto el coeficiente r_0 que se hace $r_0=\alpha$.

Así pues, una falta de anclaje en la línea i -ésima se detecta mediante cualquiera de los coeficientes espectrales de la función que sea distinto de cero. En este caso no hace falta la condición de que los términos de la PLA sean disjuntos. En la PLA de la figura 3.8, las faltas de anclaje en las salidas son detectadas por los coeficientes espectrales $r_{F_1}(100)$, $r_{F_1}(111)$, $r_{F_2}(100)$, $r_{F_2}(110)$, $r_{F_2}(011)$.

(b) Falta de anclaje en la línea de producto j -ésima: Si la línea de producto j -ésima se encuentra anclada a α , se tiene que P_j se transforma en $P_j=\alpha$. Según (3.10), se producirá un cambio en todos aquellos coeficientes que verifiquen (3.11) en las funciones a las que contribuya dicho producto.

Si, por ejemplo se produce una falta de anclaje en la línea de producto L3 de la PLA de la figura 3.8, se podrá detectar mediante los coeficientes espectrales $r_{F_1}(100)$, $r_{F_1}(110)$, $r_{F_2}(100)$, $r_{F_2}(110)$.

(c) Falta de anclaje en la línea de bit k -ésima: Una falta de anclaje en una línea de bit afectará a los productos en los que interviene. Si se supone que la línea de bit afecta al producto P_j y que la línea de bit afectada corresponde a la variable y_{σ_r} , se tiene que P_j (3.10) se transforma en

$$P_j = y_{\sigma_1} \dots \alpha (1 \oplus y_{\sigma_{r+1}}) \dots (1 \oplus y_{\sigma_s}) \quad (3.12)$$

y si la línea de bit afectada corresponde a la variable $y_{\sigma_{r+1}}$, el producto P_j se transforma en

$$P_j = y_{\sigma_1} \dots y_{\sigma_r} \alpha \dots (1 \oplus y_{\sigma_s}) \quad (3.13)$$

Así pues, si la línea de bit anclada es tal que se aplica (3.12) se producirán cambios en todos los coeficientes espectrales de la función a la que contribuya P_j y que verifiquen (3.11) y, si $\alpha=1$, también en los que verifiquen

$$\begin{aligned} w_k &= 0 \text{ si } k \neq \sigma_t \quad (t=1, \dots, s) \\ w_k &= 1 \text{ si } k = \sigma_t \quad (t=1, \dots, r-1) \end{aligned} \quad (3.14)$$

y si la línea de bit es tal que se aplica (3.13), se producirán cambios en los coeficientes espectrales que verifiquen (3.11) y además, si $\alpha=1$, en los que verifiquen

$$\begin{aligned} w_k &= 0 \text{ si } k \neq \sigma_t \quad (t=1, \dots, s, r+1) \\ w_k &= 1 \text{ si } k = \sigma_t \quad (t=1, \dots, r) \end{aligned} \quad (3.15)$$

Por ejemplo, una falta de anclaje a 1 en la línea de bit B3 de la PLA de la figura 3.8 dará lugar a cambios en los coeficientes espectrales $r_{F_1}(100)$, $r_{F_1}(110)$, $r_{F_1}(101)$, $r_{F_1}(111)$, $r_{F_2}(011)$, $r_{F_2}(001)$.

(d) Falta de contacto en el plano OR: Una falta de contacto en el plano OR de la PLA tiene como consecuencia la aparición de un término producto nuevo (contacto extra) en una de las funciones de la PLA o la desaparición de uno de sus productos (pérdida de contacto).

Si se supone que P_j es el producto que aparece o desaparece, se tiene que los coeficientes espectrales de la función afectada que cambian son los que verifican (3.11), suponiendo la expresión (3.10) para el producto P_j .

Así, si se pierde el contacto de la línea de producto L1 con F_1 en la PLA de la figura 3.8, se podrá detectar mediante los coeficientes espectrales $r_{F_1}(111)$ y $r_{F_1}(110)$.

(e) Falta de contacto en el plano AND: Si se produce una falta de pérdida de contacto en el plano AND, el efecto es similar al de una falta de línea de bit anclada a 1 en la línea de bit afectada por esa pérdida de contacto, ya que habría que eliminar la variable que corresponda a dicha línea de bit del producto P_j correspondiente a la línea de producto afectada.

En el caso de que se produzca una falta de contacto extra, el producto correspondiente a la línea de producto afectada se transforma en

$$P'_j = P_j Y_k \quad (3.16)$$

$$\text{ó} \quad P'_j = P_j Y'_k = P_j (1 \oplus Y_k) \quad (k \neq \sigma_t, \quad t=1, \dots, s) \quad (3.17)$$

Por consiguiente, en el caso de que se tenga (3.17), los coeficientes afectados por el cambio son aquellos que verifican

$$\begin{aligned} w_v &= 0 \text{ si } v \neq \sigma_t \quad (t=1, \dots, s, k) \\ w_v &= 1 \text{ si } v = \sigma_t \quad (t=1, \dots, r, k) \end{aligned} \quad (3.18)$$

y si se tiene (3.16), además de los coeficientes que verifican (3.18), también se verán afectados los que verifican

$$\begin{aligned} w_v &= 0 \text{ si } v \neq \sigma_t \quad (t=1, \dots, s) \\ w_v &= 1 \text{ si } v = \sigma_t \quad (t=1, \dots, r) \end{aligned} \quad (3.19)$$

Como ejemplo, si se produce una falta de contacto extra en la PLA de la figura 3.8 entre la línea de bit B1 y la línea de producto L2, se detectará mediante cambios en los coeficientes espectrales $r_{F_2}(111)$ y $r_{F_2}(011)$.

Con el estudio de la detección de las faltas que modelan los defectos en las PLAs, concluye el presente apartado en el que se han estudiado los cambios en los coeficientes espectrales de Tamari que se producen debido a las faltas que modelan los defectos en distintos circuitos. A partir de los resultados obtenidos, se pueden establecer de forma inmediata los coeficientes que permiten detectar ciertas faltas y con ello generar un conjunto de patrones de test. Esta información se puede utilizar para deducir cambios en coeficientes de circuitos complejos a partir de los cambios en los coeficientes de circuitos más simples. Para ello se utiliza el procedimiento modular que se detalla en la siguiente sección.

3.3. Procedimiento de selección de patrones de test para las faltas de anclaje.

En esta sección se describe un procedimiento para seleccionar un conjunto de patrones de test de las faltas de anclaje simples a partir de los coeficientes espectrales de Tamari. El procedimiento se basa en elegir un conjunto de coeficientes espectrales que puedan cambiar debido a la presencia de las faltas de anclaje simples. Si un coeficiente espectral de Tamari cambia debido a una falta, eso significa que la función que sintetiza el circuito cambia para un número impar de los valores de los que depende dicho coeficiente, determinados según la propiedad 3.2. Por consiguiente, entre los valores de los que dependa el coeficiente espectral se encuentra al menos un

patrón de test para la falta que origina el cambio en el coeficiente. Las etapas que constituyen el procedimiento son las siguientes:

1. Simulación funcional exhaustiva del circuito sin faltas.
2. Obtención de los coeficientes espectrales de Tamari para las funciones sintetizadas en cada nudo.
3. Selección de un coeficiente espectral para cada nudo entre los que pueden cambiar debido a una falta de anclaje en dicho nudo.

Las etapas 1 y 2 de este procedimiento se pueden realizar de manera conjunta ya que, como se vió en el apartado 3.1.2, es posible generar un coeficiente espectral de Tamari por cada combinación de entradas que se aplica, ordenadamente, al circuito. Por consiguiente, a medida que se van simulando combinaciones de entradas se va completando el espectro de Tamari de las funciones que sintetizan cada uno de los nudos del circuito. Además, recuérdese que para almacenar el espectro de Tamari de una función, se necesita la misma capacidad de memoria que para almacenar su tabla de verdad.

La selección de coeficientes espectrales que se indica en la etapa 3 se lleva a cabo aplicando una heurística basada en los corolarios 3.1 y 3.3. A continuación se describe dicha heurística, para un circuito de m entradas, $Y=(y_1, y_2, \dots, y_m)$, y n salidas $F=(f_1, \dots, f_n)$.

Dado un nudo del circuito, se presentan tres posibilidades: puede corresponder a una entrada del circuito, y_i , a una salida, f_j , o a un nudo interno, g .

a) Si se trata de un nudo de entrada, y_i , aplicando el corolario 3.1, se anularán todos aquellos coeficientes espectrales $r_j(\beta) \neq 0$ ($j=1, \dots, n$) con valores de $\beta=(\beta_1, \dots, \beta_m)$ tales que $\beta_i \neq 0$.

b) Si se trata de un nudo de salida, f_j , se anularán todos los coeficientes $r_j(\beta) \neq 0$, excepto $r_j(0)$ si se tiene una falta de anclaje a 1, puesto que la función en ese nudo es una constante igual al valor del anclaje.

c) Si se trata de un nudo interno, g , que sintetiza la función

$$g(Y) = \sum'_{\sigma=0}^N r_g(\sigma) y_1^{\sigma_1} \dots y_m^{\sigma_m} \quad (\sigma=(\sigma_1, \dots, \sigma_m)), \quad (3.20)$$

utilizando una función $h_j(Y)$ para cada función sintetizada por el circuito verificando que $f_j(Y) = h_j(Y, g(Y))$, para todo $Y \in C_2^m$ (teorema 2.3) se tiene que

$$f_j(Y) = \sum'_{\alpha=0}^N r_{j,1}(\alpha) y_1^{\alpha_1} \dots y_m^{\alpha_m} \oplus g\{\sum'_{\beta=0}^N r_{j,2}(\beta) y_1^{\beta_1} \dots y_m^{\beta_m}\} \quad (3.21)$$

y sustituyendo en (3.20) en (3.21) se obtiene

$$f_j(Y) = \sum_{\alpha=0}^N r_{j,1}(\alpha) Y_1^{\alpha^1} \dots Y_m^{\alpha^m} \oplus \left\{ \sum_{\sigma=0}^N r_g(\sigma) Y_1^{\sigma^1} \dots Y_m^{\sigma^m} \right\} \left\{ \sum_{\beta=0}^N r_{j,2}(\beta) Y_1^{\beta^1} \dots Y_m^{\beta^m} \right\} \quad (3.22)$$

A partir de (3.22) y utilizando la propiedad 3.4 se determina la dependencia entre los coeficientes espectrales de la función f_j y los de la función que sintetiza el nudo g :

$$r_j(\alpha) = r_{j,1}(\alpha) \oplus \sum_{\beta \vee \sigma = \alpha} r_g(\sigma) r_{j,2}(\beta) \quad (3.23)$$

Por consiguiente, como una falta de anclaje hace que se anulen todos los coeficientes espectrales $r_g(\sigma) \neq 0$ con $\sigma \neq 0$, los coeficientes espectrales de la función f_j que podrían detectar una falta en dicho nudo se encuentran entre los $r_j(\alpha)$ para los que $\sigma \subseteq \alpha$.

De esta forma, teniendo en cuenta a), b) y c), para seleccionar un conjunto de coeficientes espectrales que cambien debido a las faltas de anclaje simples y, por consiguiente, definan un conjunto de patrones de test para dichas faltas, se sigue el proceso que se detalla a continuación.

Algoritmo 3.2.

1. Para cada nudo del circuito, g :

1.1. Seleccionar un coeficiente espectral de la función sintetizada por el nudo g , $r_g(\sigma) \neq 0$.

1.2. Marcar todos los coeficientes espectrales que en alguna de las funciones, f_1, \dots, f_m , verifican $r_j(\alpha) \neq 0$ con $\sigma \subseteq \alpha$.

2. Seleccionar entre los coeficientes espectrales marcados en 1.2 aquél $r_j(\alpha)$ para el que el número de bits en el índice α es mínimo.

3. Eliminar todos aquellos nudos que en 1.2 han marcado el coeficiente espectral seleccionado.

4. Comprobar si queda algún nudo por considerar. En caso afirmativo, repetir el paso 2; si no, terminar. ■

En el paso 2 del algoritmo se seleccionan los coeficientes de índices con un número mínimo de unos ya que cuanto menor es dicho número menos patrones de test se añaden al test, según la propiedad 3.2.

El número de veces que hay que repetir los pasos 1.1 y 1.2 del algoritmo 3.2 es proporcional al número de nudos, igualmente también es proporcional al número de nudos el número de veces que se han de repetir 2, 3 y 4, en el peor de los casos. La complejidad de los pasos 1.1, 1.2, 2 y 3 depende del número de coeficientes espectrales a considerar, es decir, crece exponencialmente con

el número de entradas en el peor de los casos. Consiguientemente, el límite de aplicabilidad de este procedimiento de selección de patrones viene determinado más por el incremento en el número de entradas del circuito que por el incremento del número de nudos. Eso es debido a que no es necesario simular el circuito para cada una de las entradas y para cada una de las faltas de anclaje posibles: la información del comportamiento del circuito se obtiene a partir de una única simulación, mediante los coeficientes espectrales.

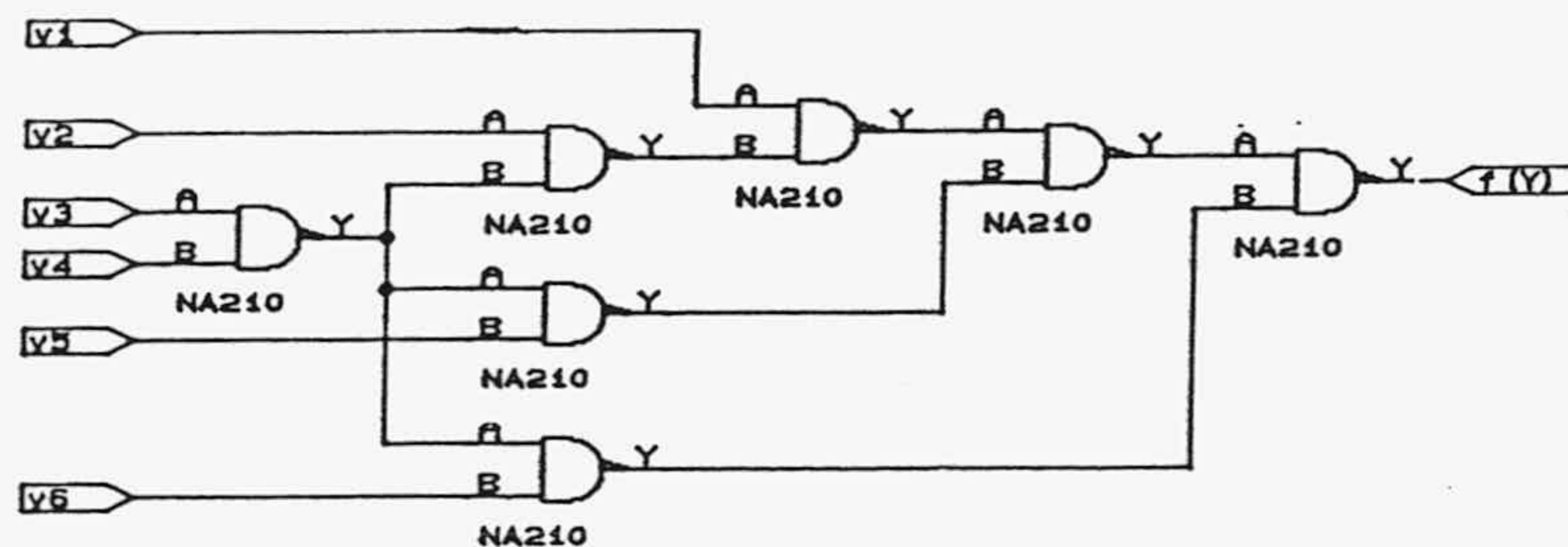


Figura 3.9. Ejemplo de circuito con reconvergencias.

Como ejemplo de aplicación de este procedimiento de selección de patrones considérese el circuito de la figura 3.9. Se trata de un circuito utilizado en [SAV84] para ilustrar la aplicación del algoritmo de Corte, que allí se describe, y constituye un ejemplo de circuito con reconvergencias. Se trata de un circuito con 6 entradas y 13 nudos, de los cuales 6 son nudos internos y uno de ellos es un nudo de reconvergencia que actúa sobre tres puertas lógicas.

El resultado de la simulación funcional del circuito y del cálculo de los coeficientes espectrales se muestra en la Tabla 3.1, donde aparecen aquellos coeficientes espectrales que son distintos de cero para las funciones que sintetizan los nudos del circuito. En dicha Tabla, el coeficiente $C(i)$ corresponde al término $y_6^{i_6} \dots y_1^{i_1}$, siendo $i = i_6 + i_5 \cdot 2 + \dots + i_1 \cdot 2^5$. En la Tabla 3.2 se han marcado con una cruz los coeficientes espectrales de la función, $f(Y)$, candidatos a cambiar debido a alguna falta de anclaje en alguno de los nudos según el paso 1.2 del algoritmo 3.2.

Como resultado de la aplicación del paso 2 del algoritmo 3.2, a partir de la Tabla 3.2, los coeficientes se seleccionarán de la siguiente forma:

- para N_0 , $C(2) = y_5$, y se eliminan los nudos N_0 , N_5 , $f(Y)$, y_5 ;
- para N_1 , $C(14) = y_3 y_4 y_5$, y se eliminan N_1 , y_3 , y_4 ;
- para N_2 , $C(33) = y_6 y_1$, y se eliminan N_2 , N_4 , y_1 , y_6 ;
- para N_3 , $C(48) = y_1 y_2$, y se eliminan N_3 , y_2 .



Tabla 3.1. Coeficientes espectrales para las funciones sintetizadas por cada nodo del circuito.

	f(Y)	y6	y5	y4	y3	y2	y1	N0	N1	N2	N3	N4	N5
C(0)	1	0	0	0	0	0	0	1	1	1	1	1	1
C(1)	0	1	0	0	0	0	0	0	0	1	0	0	0
C(2)	1	0	1	0	0	0	0	1	0	0	0	0	1
C(3)	1	0	0	0	0	0	0	0	0	0	0	0	0
C(4)	0	0	0	1	0	0	0	0	0	0	0	0	0
C(8)	0	0	0	0	1	0	0	0	0	0	0	0	0
C(12)	0	0	0	0	0	0	0	0	1	0	0	0	0
C(13)	0	0	0	0	0	0	0	0	0	1	0	0	0
C(14)	1	0	0	0	0	0	0	1	0	0	0	0	1
C(15)	1	0	0	0	0	0	0	0	0	0	0	0	0
C(16)	0	0	0	0	0	1	0	0	0	0	1	0	0
C(28)	0	0	0	0	0	0	0	0	0	0	1	0	0
C(32)	1	0	0	0	0	0	1	0	0	0	0	1	1
C(33)	1	0	0	0	0	0	0	0	0	0	0	0	0
C(34)	1	0	0	0	0	0	0	0	0	0	0	0	1
C(35)	1	0	0	0	0	0	0	0	0	0	0	0	0
C(45)	1	0	0	0	0	0	0	0	0	0	0	0	0
C(46)	1	0	0	0	0	0	0	0	0	0	0	0	1
C(47)	1	0	0	0	0	0	0	0	0	0	0	0	0
C(48)	1	0	0	0	0	0	0	0	0	0	0	1	1
C(49)	1	0	0	0	0	0	0	0	0	0	0	0	0
C(50)	1	0	0	0	0	0	0	0	0	0	0	0	1
C(51)	1	0	0	0	0	0	0	0	0	0	0	0	0
C(60)	1	0	0	0	0	0	0	0	0	0	0	1	1
C(61)	1	0	0	0	0	0	0	0	0	0	0	0	0
C(62)	1	0	0	0	0	0	0	0	0	0	0	0	1
C(63)	1	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 3.2. Coeficientes espectrales de f marcados para cada nodo

	f(Y)	y6	y5	y4	y3	y2	y1	N0	N1	N2	N3	N4	N5
C(2)	X		X					X					X
C(3)	X	X	X					X		X			X
C(14)	X		X	X	X			X	X				X
C(15)	X	X	X	X	X			X	X	X			X
C(32)	X						X					X	X
C(33)	X	X					X			X		X	X
C(34)	X		X				X	X				X	X
C(35)	X	X	X				X	X		X		X	X
C(45)	X	X		X	X		X		X	X		X	X
C(46)	X		X	-X	X		X	X	X			X	X
C(47)	X	X	X	X	X		X	X	X	X		X	X
C(48)	X					X	X				X	X	X
C(49)	X	X				X	X			X	X	X	X
C(50)	X		X			X	X	X		X	X	X	X
C(51)	X	X	X			X	X	X		X	X	X	X
C(60)	X			X	X	X	X		X		X	X	X
C(61)	X	X		X	X	X	X		X	X	X	X	X
C(62)	X		X	X	X	X	X	X	X		X	X	X
C(63)	X	X	X	X	X	X	X	X	X	X	X	X	X

```

P y6,y5,y4,y3,y2,y1
DATS
000000
010000
000010
000001
000011
100000
100001
000100
001000
001100
010100
011000
011100
DATE
SETV CUENTA,1
AL SDC CUENTA
ST DATA (y6,y5,y4,y3,y2,y1)
SU TIME=**+1000
INCR CUENTA,1
IFV CUENTA < 14 GT AL
F
    
```

Figura 3.10. Fichero de patrones de test (entre DATS y DATE)

En la figura 3.10 se muestra el fichero de estímulos generado. Este fichero se utiliza como entrada para el simulador de faltas del LESIM2. De esta manera se puede obtener el cubrimiento de faltas que proporciona el conjunto de patrones seleccionado. En la figura 3.11 se muestra el incremento del cubrimiento de faltas a medida que se van aplicando los patrones de test para dos conjuntos de patrones. En línea discontinua, se muestra el cubrimiento para un conjunto de patrones seleccionados aleatoriamente y en línea continua los correspondientes al procedimiento aquí presentado. Se observa que, aunque en la selección aleatoria, con pocos patrones se consigue un cubrimiento elevado, se estanca y no se alcanza un cubrimiento del 100% para el mismo número de patrones que los que componen el test determinista obtenido. Esta situación indica que podría ser ventajoso utilizar un conjunto de patrones seleccionados aleatoriamente hasta conseguir determinado nivel de cubrimiento y completar con patrones seleccionados de forma determinista para las faltas no detectadas.

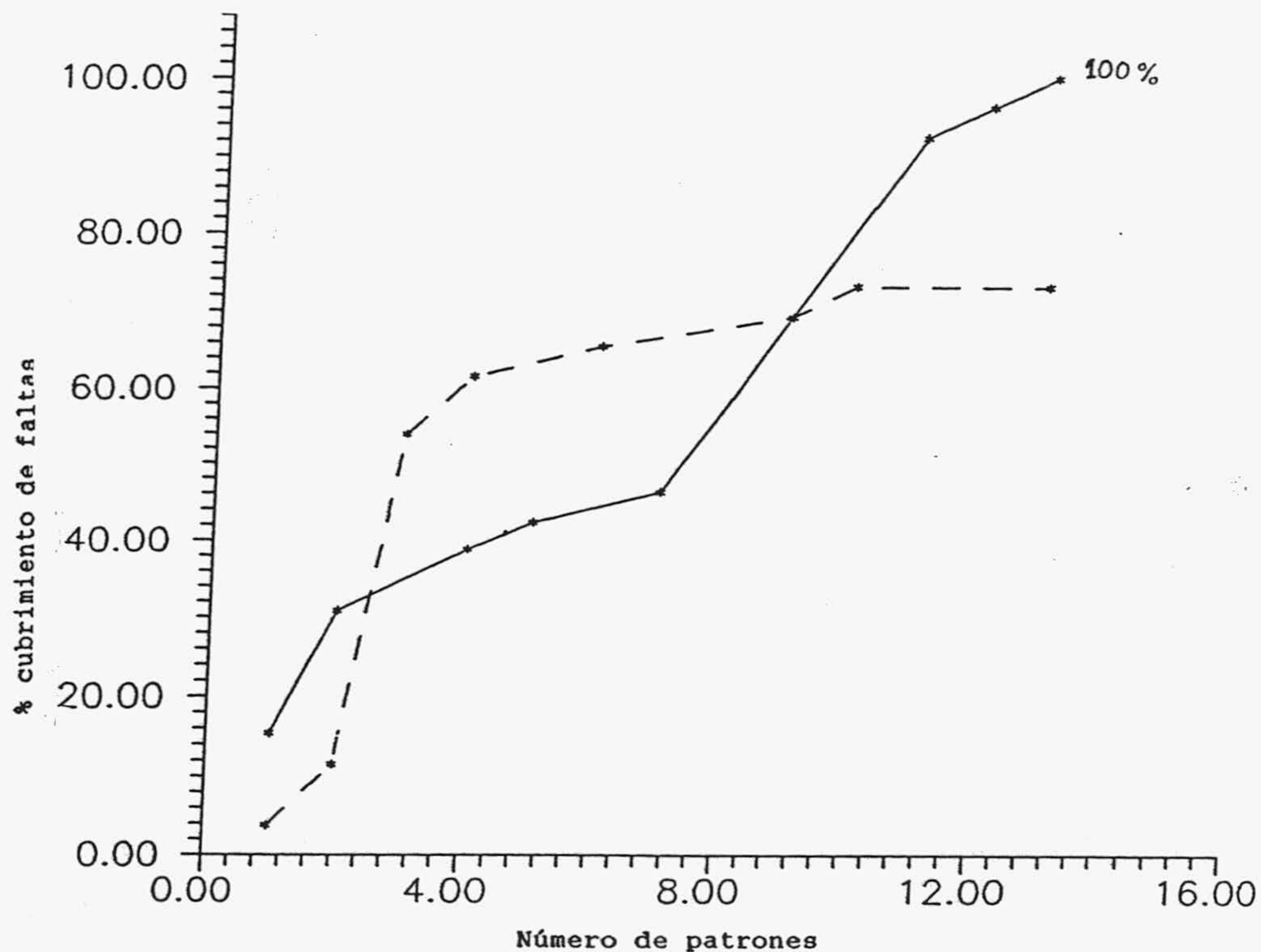


Figura 3.11. Cubrimiento de faltas frente a patrones aplicados

La Tabla 3.3 indica, mediante un asterisco (*), qué coeficientes espectrales de f cambian debido a las faltas de anclaje a 1 en cada nudo. Como se puede comprobar, coinciden con los coeficientes marcados que se dan en la Tabla 3.2, salvo en las reconvergencias. Para éstas, el conjunto de

coeficientes marcados, es decir los coeficientes marcados para N1, contienen al conjunto de los coeficientes espectrales marcados. Como se puede comprobar, los coeficientes espectrales seleccionados mediante el procedimiento propuesto no están entre los coeficientes que cambian para esas faltas. Esto no significa que los patrones obtenidos a partir de tales coeficientes no puedan detectar esas faltas. En el caso de que no se detectara una falta para determinado nudo, se volvería a repetir el procedimiento para ese nudo seleccionando entre los coeficientes espectrales que estén marcados para ese nudo y que no hayan sido seleccionados ya.

Tabla 3.3. Coeficientes que cambian para las faltas de anclaje en cada nudo.

	f(Y)	y6	y5	y4	y3	y2	y1	N0	N1	N2	N3	N4	N5	N1.1	N1.2	N1.3
C(2)	*		*					*					*			
C(3)	*	*	*					*		*			*			
C(14)	*		*	*	*			*	*				*	*	X	X
C(15)	*	*	*	*	*			*	*	*			*	X	X	X
C(32)	*						*					*	*			
C(33)	*	*					*			*		*	*			
C(34)	*		*				*	*				*	*			
C(35)	*	*	*				*	*		*		*	*			
C(45)	*	*		*	*		*		*	*		*		X	X	*
C(46)	*		*	*	*		*	*	*			*	*	*	X	X
C(47)	*	*	*	*	*		*	*	*	*		*	*	X	X	X
C(48)	*					*	*				*	*	*			
C(49)	*	*				*	*			*	*	*	*			
C(50)	*		*			*	*	*			*	*	*			
C(51)	*	*	*			*	*	*		*	*	*	*			
C(60)	*			*	*	*	*		*		*	*	*	X	*	X
C(61)	*	*		*	*	*	*		*	*	*	*	*	X	X	X
C(62)	*		*	*	*	*	*	*	*		*	*	*	X	X	X
C(63)	*	*	*	*	*	*	*	*	*	*	*	*	*	X	X	X

3.4. Conclusión.

En el presente capítulo se han particularizado a circuitos binarios los resultados del capítulo 2, obteniéndose una serie de conclusiones que permiten la aplicación de las técnicas espectrales basadas en la transformada de Tamari al test de circuitos integrados. Se ha elaborado un procedimiento de selección de patrones de test utilizando la información obtenida respecto a los cambios en los coeficientes espectrales debidos a las faltas de anclaje simples. Dicho procedimiento no precisa de la simulación del circuito para cada una de las posibles faltas de anclaje.

El siguiente capítulo se dedica a describir un procedimiento modular para el test de circuitos complejos mediante el empleo de coeficientes espectrales.

CAPITULO 4. PROCEDIMIENTO MODULAR PARA EL TEST ESPECTRAL.

Los métodos de generación de patrones o de compresión de respuestas basados en procedimientos espectrales deben permitir la predicción de los coeficientes que cambien al producirse cualquiera de los defectos que se pretenda detectar. Los cambios previsibles en los coeficientes espectrales en el caso de defectos modelados como faltas de anclaje simple se deducen de los corolarios 3.1 a 3.4. Teniéndolos en cuenta, para determinar si cambia un coeficiente espectral en una función de n variables debido a una falta de anclaje en una línea interna es preciso calcular coeficientes espectrales para funciones de n y de $n+1$ variables en el caso general (corolario 3.2). Como la complejidad de cálculo de los coeficientes espectrales de Tamari aumenta exponencialmente con el número de entradas al circuito y puesto que el número de nudos y de entradas del circuito puede ser muy elevado, el tiempo necesario para obtener los coeficientes de test será excesivo a no ser que se disponga de un procedimiento para relacionar los coeficientes espectrales de partes más simples del circuito con los del circuito completo.

En este capítulo se describe un procedimiento modular para el cálculo del espectro de Tamari de la función que realiza un circuito a partir de los coeficientes espectrales de los módulos que lo constituyen. Mediante este procedimiento se puede reducir la complejidad de obtención de coeficientes espectrales en circuitos con muchas entradas. Además, a diferencia de lo que ocurre en [DAR89a], aquí se consideran las faltas de anclaje en nudos internos de circuitos sin restricciones en su topología. En una primera sección se

describe un procedimiento modular para el cálculo de los coeficientes espectrales y se presenta un algoritmo que implementa dicho procedimiento. En la sección 4.2 se consideran los cambios en los coeficientes espectrales originales por las faltas de anclaje simples. A partir de los contenidos de las secciones 4.1 y 4.2 se ha elaborado un procedimiento para la selección de patrones, que se presenta en la sección 4.3. Las dos últimas secciones presentan una aplicación del procedimiento para circuitos en los que los módulos estén interconectados como un array lineal.

4.1. Procedimiento modular para el cálculo de la transformada de Tamari.

En este apartado se pretende relacionar los cambios que los defectos ocasionan en los coeficientes espectrales de submódulos del circuito a testear con los coeficientes espectrales de todo el circuito.

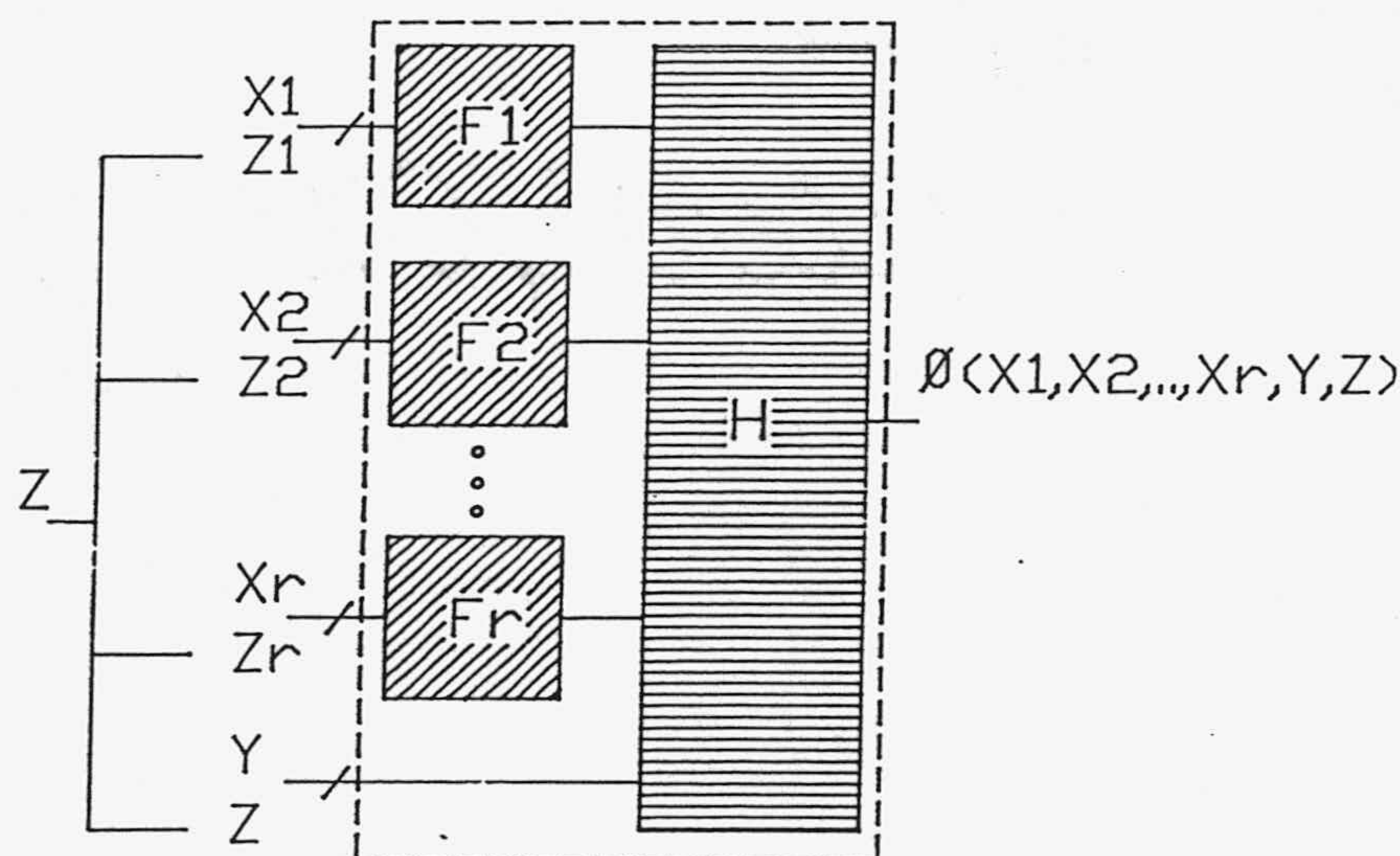


Figura 4.1. Partición general de un circuito lógico.

Dado un circuito, siempre es posible llevar a cabo una partición del mismo en módulos más simples como muestra la figura 4.1. En ella, se distinguen dos niveles: el de los módulos F_1, F_2, \dots, F_r , que realizan las funciones f_1, f_2, \dots, f_r , respectivamente, y el segundo nivel con el módulo H , que realiza la función h . Las variables se agrupan y se notan como sigue:

- X_i ($i=1, 2, \dots, r$) designa al conjunto de variables $(x_{i1}, x_{i2}, \dots, x_{is(i)})$ que únicamente actúan sobre el módulo F_i .
- Y designa al conjunto de variables (y_1, \dots, y_s) que sólo actúan sobre el módulo H .

- Z designa al conjunto de variables (z_1, \dots, z_t) que entran a varios módulos.

El número de variables del circuito es $\sum_1 s(i) + s + t$ ($i=1, \dots, r$) y la función sintetizada por el circuito se notará como $\phi(X_1, \dots, X_r, Y, Z)$. Los módulos F_i ($i=1, 2, \dots, r$) pueden tener partes comunes, como se puede comprobar en la partición de la figura 4.3 para el circuito de la figura 4.2, donde la puerta OR BNN1 es compartida por el módulo F_1 y por el F_2 .

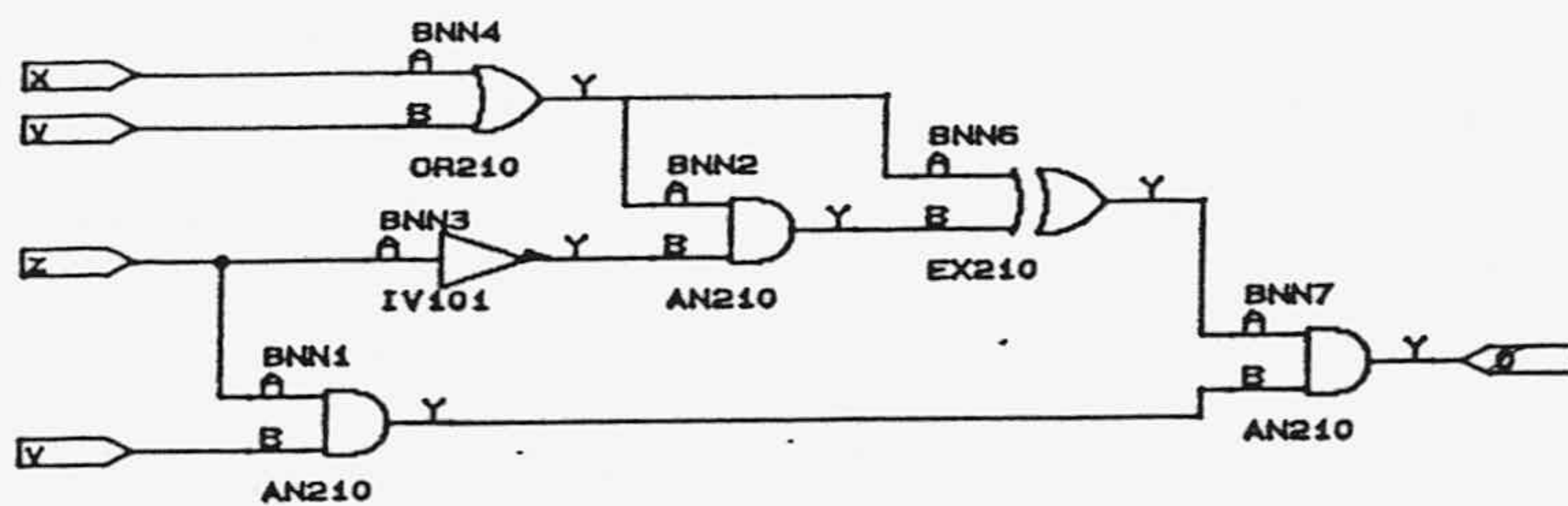


Figura 4.2. Ejemplo de circuito lógico.

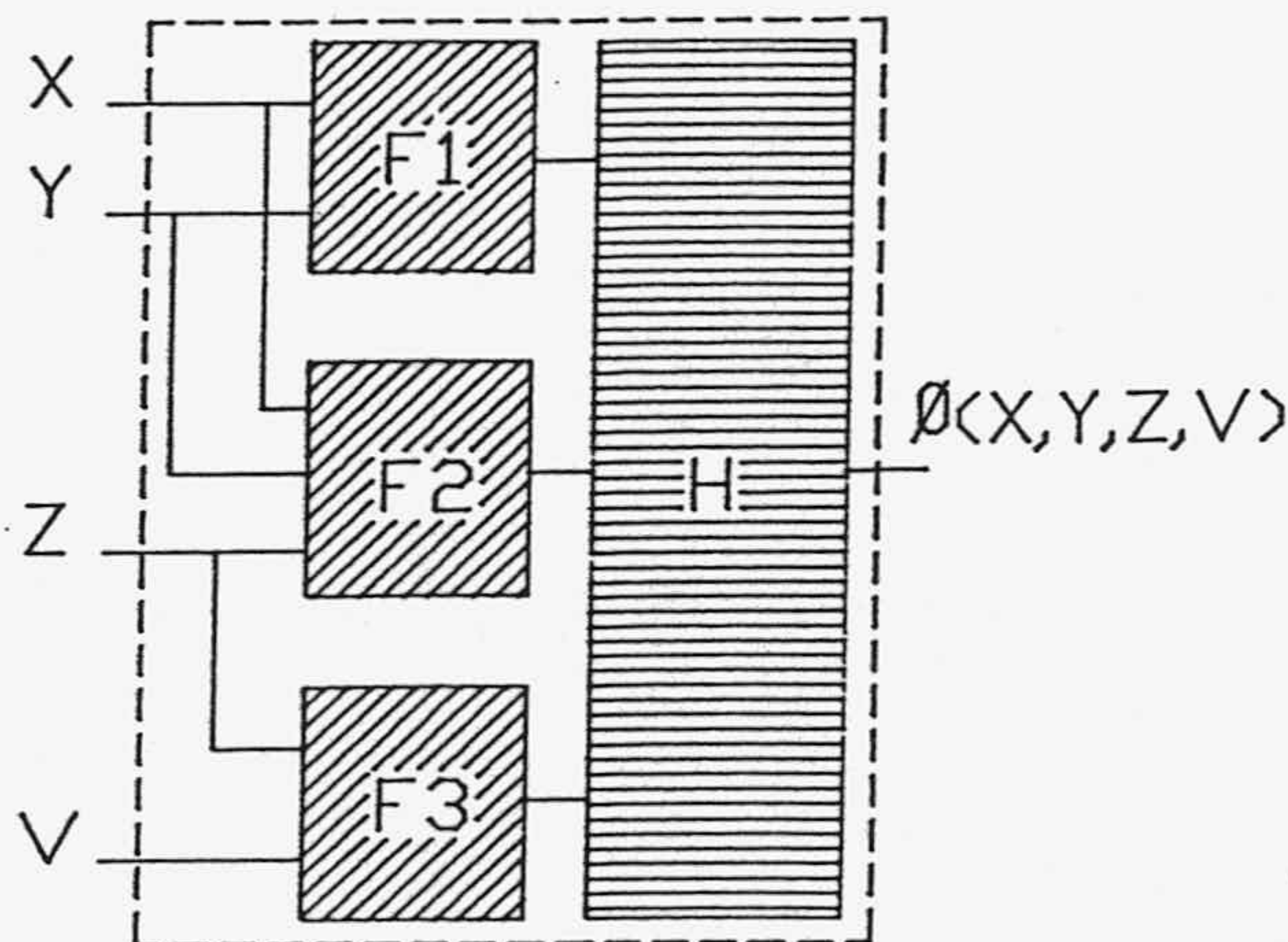
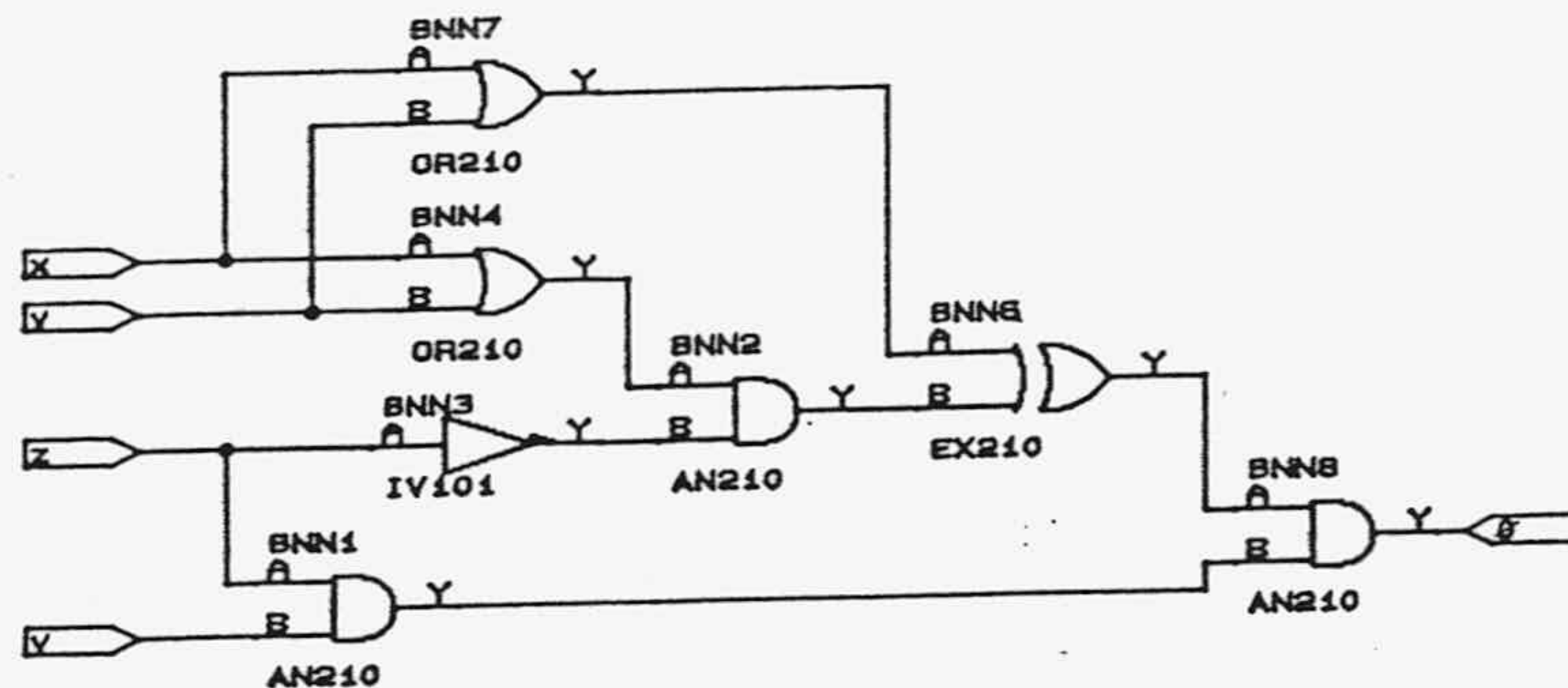


Figura 4.3. Partición del circuito de la figura 4.2.

En la partición de la figura 4.3 se pueden distinguir tres módulos F_i ($i=1, 2, 3$) y un módulo H. Las funciones que sintetizan estos módulos son:

$$\begin{aligned}
f_1(x, y) &= x \oplus y \oplus xy \\
f_2(x, y, z) &= x \oplus y \oplus xy \oplus xz \oplus yz \oplus xyz \\
f_3(z, v) &= zv \\
h(f_1, f_2, f_3) &= f_1 f_3 \oplus f_2 f_3 \\
\phi(x, y, z, v) &= xzv \oplus yzv \oplus xyzv
\end{aligned}$$

y los conjuntos de variables X , Y y Z definidos estarán constituidos de la siguiente manera: $X_1 = \emptyset$, $X_2 = \emptyset$, $X_3 = \{v\}$, $Y = \emptyset$ y $Z = (x, y, z)$ En lo que sigue, con la potencia de un vector, por ejemplo Y^α , notaremos el producto de sus componentes elevadas al índice que se obtiene a partir de la representación binaria de la potencia, es decir, $y_1^{\alpha_1} \dots y_s^{\alpha_s}$, siendo $Y = (y_1, \dots, y_s)$ y $\alpha = \alpha_1 + \dots + 2^{s-1} \alpha_s$. Así pues, las representaciones de las funciones sintetizadas por los módulos del circuito en términos de las funciones de Tamari serán:

$$f_1(X_1, Z) = \sum'_{w_1, \sigma_1} r_1(w_1, \sigma_1) X_1^{w_1} Z^{\sigma_1} \quad (4.1)$$

$$h(f_1, \dots, f_r, Y, Z) = \sum'_{\alpha, \beta, \Gamma} r_h(\alpha, \beta, \Gamma) (\prod_1 f_1^{\alpha_1}) Y^\beta Z^\Gamma \quad (4.2)$$

donde $\alpha = (\alpha_1, \dots, \alpha_r)$, $\alpha_i \in \{0, 1\}$, $w_1 \in C_2^{s(i)}$, $\sigma_1 \in C_2^t$, $\beta \in C_2^s$, $\Gamma \in C_2^t$

y en lo que se refiere a la función sintetizada por todo el circuito, $\phi(X_1, \dots, X_r, Y, Z)$, expresada en términos de las funciones de Tamari es

$$\phi(X_1, \dots, X_r, Y, Z) = \sum'_{a_1, \dots, a_r, b, c} r_\phi(a_1, \dots, a_r, b, c) X_1^{a_1} \dots X_r^{a_r} Y^b Z^c \quad (4.3)$$

donde $a_i \in C_2^{s(i)}$, $b \in C_2^s$, $c \in C_2^t$.

Sustituyendo (4.1) en (4.2) e igualando con (4.3) se tiene que:

$$r_\phi(a_1, \dots, a_r, b, c) = \sum'_{\alpha, \Gamma, w, \sigma} [r_h(\alpha, \beta, \Gamma) \{ \prod_1 r_1^{\alpha_1}(w_1, \sigma_1) \}] \quad (4.4)$$

con las condiciones $a_i = \alpha_i w_i$; $b = \beta$; $c = \Gamma \prod_1 \alpha_i \sigma_i$

Los valores a los que se extiende la sumatoria de la expresión (4.4), $\alpha_1, \dots, \alpha_r, \beta, \Gamma, \sigma$, dependen de los índices del coeficiente espectral considerado, $r_\phi(a_1, \dots, a_r, b, c)$. Se pueden obtener teniendo en cuenta lo siguiente:

$$\text{Si } a_i = 0 \quad (\alpha_i = 1, w_i = 0) \quad \text{ó} \quad (\alpha_i = 0) \quad (4.5a)$$

$$\text{Si } a_i \neq 0 \quad (\alpha_i = 1, w_i = a_i) \quad (4.5b)$$

$$\beta = b \quad (4.5c)$$

$$\text{Si } c = 0 \quad \Gamma = 0 \text{ y } (\alpha_i = 0 \text{ ó } (\alpha_i \neq 0, \sigma_i = 0)) \quad (4.5d)$$

$$\text{Si } c \neq 0 \quad (\Gamma, \sigma_i / c = \Gamma \prod_1 \alpha_i \neq 0 \sigma_i) \quad (i=1, \dots, r) \quad (4.5e)$$

Teniendo en cuenta (4.5) es posible obtener los coeficientes espectrales de un circuito a partir de los coeficientes de sus módulos constituyentes, más simples. Además, se pueden establecer las propiedades que han de cumplir los

coeficientes espectrales de la función que realiza un circuito para detectar los defectos modelados como faltas de anclaje, a partir de que se conozcan los coeficientes espectrales de los módulos que han cambiado. El algoritmo que se ha utilizado para calcular los coeficientes espectrales de la función sintetizada por todo el circuito a partir de los coeficientes de los bloques F_1, \dots, F_r y H es el siguiente:

Algoritmo 4.1.

1. Para cada coeficiente $r_h(\alpha, \beta, \Gamma) \neq 0$:

1.1. Tomar los coeficientes $r_i(w_i, \sigma_i) \neq 0$ de los módulos F_i con índices i tales que $\alpha_i \neq 0$ en $r_h(\alpha, \beta, \Gamma)$. Se supondrá que $\alpha_i \neq 0$ para $i=1, \dots, s$.

1.2. Obtener todas las combinaciones posibles de los coeficientes seleccionados en 1.1 de manera que aparezca en cada una de ellas un coeficiente espectral de cada módulo F_i con $\alpha_i \neq 0$ en $r_h(\alpha, \beta, \Gamma)$: $(r_1(w_1, \sigma_1), \dots, r_s(w_s, \sigma_s))$

1.3. Cada combinación $(r_1(w_1, \sigma_1), \dots, r_s(w_s, \sigma_s))$ indica que debe añadirse 1 (suma módulo-2) al coeficiente espectral $r_g(a_1, \dots, a_r, b, c)$ con $b=\beta$; $a_i=w_i$, si $i=1, \dots, s$; y $c=\Gamma \prod_{i=1}^s \sigma_i$.

■

La complejidad máxima para este procedimiento, suponiendo un número de coeficientes N_H distintos de cero para el módulo H y un número de coeficientes N_i para cada módulo F_i , es igual a $N_H N_1 \dots N_r \leq N^{r+1}$, donde N es el valor máximo de N_H, N_1, \dots, N_r : se trata de una complejidad polinómica respecto del número de coeficientes espectrales distintos de cero de los módulos del circuito.

A continuación se presentan algunos ejemplos de circuitos que ilustran la forma de definir módulos para aplicar (4.5). Sea de nuevo el circuito de la figura 4.3 que por su simplicidad permite comprobar a mano los valores de los coeficientes espectrales. Teniendo en cuenta los conjuntos de variables X, Y y Z para la partición realizada, los índices de los coeficientes espectrales son

$$\begin{array}{ll} r_g(a_3, c) & \text{con } c=(c_1, c_2, c_3); c_j, a_3 \in C_2 \\ r_h(\alpha_1, \alpha_2, \alpha_3) & \text{con } \alpha_j \in C_2 \\ r_1(\sigma_1) & \text{con } \sigma_1=(\sigma_{11}, \sigma_{12}); \sigma_{1j} \in C_2 \\ r_2(\sigma_2) & \text{con } \sigma_2=(\sigma_{21}, \sigma_{22}, \sigma_{23}); \sigma_{1j} \in C_2 \\ r_3(w_3, \sigma_3) & \text{con } \sigma_3=(\sigma_{33}); \sigma_3 \in C_2 \end{array}$$

verificandose $a_3=\alpha_3 w_3$; $c_1=\alpha_1 \sigma_{11} \cup \alpha_2 \sigma_{21}$; $c_2=\alpha_1 \sigma_{12} \cup \alpha_2 \sigma_{22}$; $c_3=\alpha_2 \sigma_{23} \cup \alpha_3 \sigma_{33}$.

A continuación, se procede al cálculo del coeficiente espectral $r_g(1,1,0,1)$ a partir de (4.3), (4.4) y (4.5). Los índices de este coeficiente espectral son $a_3=1$, $c_1=1$, $c_2=0$ y $c_3=1$. En primer lugar, aplicando (4.5a),

puesto que $a_3=1$ se tiene que $\alpha_3=1$ y $w_3=1$. De entre los coeficientes espectrales de la función h con índice $\alpha_3=1$, sólo $r_h(1,0,1)$ y $r_h(0,1,1)$ son distintos de cero, por lo que las únicas posibilidades para los índices α son $(\alpha_1=1, \alpha_2=0, \alpha_3=1)$ y $(\alpha_1=0, \alpha_2=1, \alpha_3=1)$.

Los valores de los índices espectrales σ que intervienen en el cálculo de $r_\sigma(1,1,0,1)$ se determinarán a partir de (4.5e) puesto que $c \neq 0$. Para $(\alpha_1, \alpha_2, \alpha_3)=(1,0,1)$ los valores de σ que hay que obtener son σ_1 y σ_3 , que según (4.5e) han de verificar $c_1=1=\sigma_{11}$, $c_2=0=\sigma_{12}$, $c_3=1=\sigma_{33}$, por lo que resulta $\sigma_1=(1,0)$ $\sigma_3=(1)$.

En el caso de $(\alpha_1, \alpha_2, \alpha_3)=(0,1,1)$ los valores de σ a obtener son σ_2 y σ_3 . Utilizando nuevamente (4.5e) se tiene que los valores posibles para σ_2 y σ_3 son

$$\begin{aligned} \sigma_2 &= (1,0,0) & ; & \sigma_3 = (1) \\ \sigma_2 &= (1,0,1) & ; & \sigma_3 = (1) \\ \sigma_2 &= (1,0,1) & ; & \sigma_3 = (0) \end{aligned}$$

Así, aplicando (4.4), el valor del coeficiente que se quiere calcular se obtiene a partir de

$$\begin{aligned} r_\sigma(1,0,1,1) &= r_h(1,0,1)r_1(1,0)r_3(1,1) \oplus r_h(0,1,1)[r_2(1,0,1)r_3(1,1) \oplus \\ &\oplus r_2(1,0,0)r_3(1,1) \oplus r_2(1,0,1)r_3(1,0)] = 1 \end{aligned}$$

que, como se puede comprobar a partir de la expresión de ϕ , es correcto, puesto que en su desarrollo aparece el término vxz .

Otro ejemplo es el circuito cuya partición en bloques se muestra en la figura 4.4. Se trata de un circuito con seis entradas $(x_1, x_2, x_3, y, z_1, z_2)$ en el que se pueden distinguir tres módulos, F_1 , F_2 y H , que sintetizan, respectivamente, las funciones $f_1(x_1, x_2, z_1)$, $f_2(x_3, z_1, z_2)$ y $h(f_1, f_2, y, z_2)$. En esta partición las agrupaciones de variables son $X_1=(x_1, x_2)$, $X_2=(x_3)$, $Z=(z_1, z_2)$ e $Y=(y)$

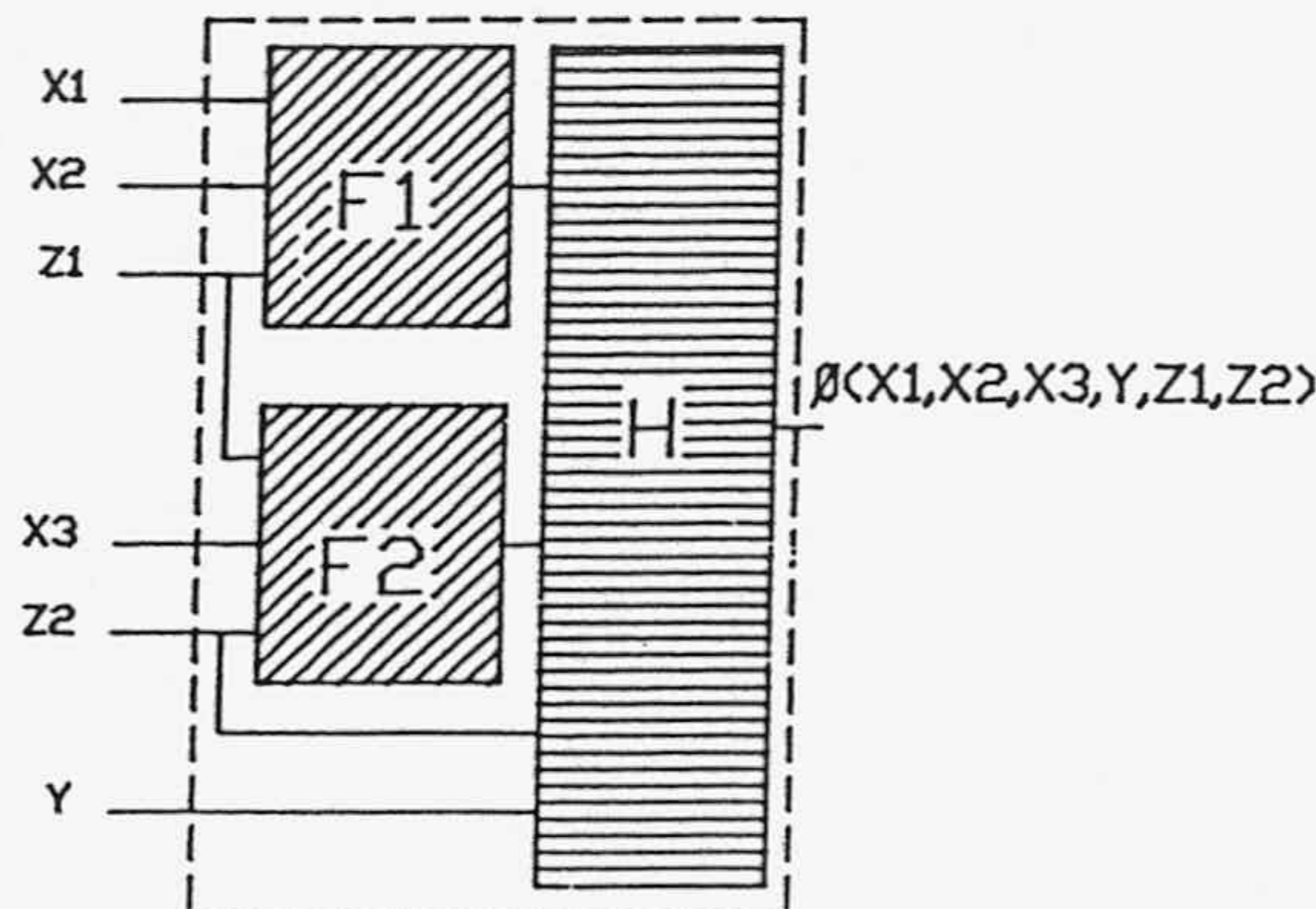


Figura 4.4. Ejemplo de partición de un circuito para aplicar el procedimiento modular.

Para determinar el coeficiente $r_g(1,0,1,1,0,1)$ de la función $\phi(x_1, x_2, x_3, y, z_1, z_2)$ se puede utilizar (4.4). Como $a_1=(1,0)$, $a_2=(1)$, $b=(1)$ y $c=(0,1)$, los índices de la sumatoria deben cumplir $\alpha_1 w_1 = a_1 = (1,0)$, $\alpha_2 w_2 = a_2 = (1)$, $\Gamma \cup \alpha_1 \sigma_1 \cup \alpha_2 \sigma_2 = c = (0,1)$, $\beta = b = 1$. Es decir, $\alpha_1 = 1$, $w_1 = (1,0)$, $\alpha_2 = 1$, $w_2 = (1)$, $\sigma_1 = (0,0)$ y los valores de Γ y σ_2 acoplados, siendo posibles las parejas $(\sigma_2 = (0,1), \Gamma = (0,0))$, $(\sigma_2 = (0,1), \Gamma = (0,1))$ y $(\sigma_2 = (0,1), \Gamma = (0,1))$. Con esto se tiene que

$$r_g(1,0,1,1,0,1) = r_h(1,1,1,1) r_1(1,0,0) (r_2(1,0,0) \oplus r_2(1,0,1)) \oplus \\ \oplus r_h(1,1,1,0) r_1(1,0,0) r_2(1,0,1).$$

El coeficiente $r_g(1,0,1,1,0,1)$ del circuito de la figura 4.4, como cualquier otro, también se puede determinar utilizando las condiciones (4.5). Así:

$$\begin{array}{lll} a_1 = (1,0) \neq 0 & \text{implica que} & \alpha_1 = 1, w_1 = a_1 = (1,0) \\ a_2 = (1) \neq 0 & \text{implica que} & \alpha_2 = 1, w_2 = a_2 = (1) \\ b = 1 & \text{implica que} & \beta = 1 \end{array}$$

Según esto, los coeficientes que participan en la sumatoria tienen la forma:

$$r_1(1,0,\sigma_{1,1}) \quad r_2(1,\sigma_2) \quad r_h(1,1,1,\Gamma_2)$$

donde $\sigma_1 = (\sigma_{1,1}, 0)$, $\sigma_2 = (\sigma_{2,1}, \sigma_{2,2})$ y $\Gamma = (\Gamma_1, \Gamma_2)$, es decir $\sigma_{1,2} = 0$ y $\Gamma_1 = 0$ teniendo en cuenta las conexiones de las variables z_1 y z_2 a los distintos módulos. Por último queda fijar los valores de los σ y de Γ :

$$c = (0,1) \text{ implica que } \Gamma_1 = 0, \sigma_{1,1} = 0, \sigma_{2,1} = 0 \text{ y } \sigma_{2,2} \cup \Gamma_2 = 1$$

Es decir que en cada término de la sumatoria módulo-2 intervienen los productos $r_1(1,0,0) r_2(1,0,\sigma_{2,2}) r_h(1,1,1,\Gamma_2)$ siempre que verifiquen que $\sigma_{2,2} \cup \Gamma_2 = 1$:

$$\begin{array}{l} r_1(1,0,0) r_2(1,0,0) r_h(1,1,1,1) \\ r_1(1,0,0) r_2(1,0,1) r_h(1,1,1,0) \\ r_1(1,0,0) r_2(1,0,1) r_h(1,1,1,1) \end{array}$$

como se había obtenido anteriormente.

4.2. Cambios en los coeficientes espectrales.

En esta sección se establecen los criterios que permiten la predicción de los coeficientes espectrales que cambian en la función que realiza un circuito cuando se produce una falta de anclaje y se conocen los cambios que dicha falta ocasiona en los coeficientes espectrales de los módulos constituyentes del circuito.

(a) Si una falta de anclaje simple en el módulo F_1 modifica el coeficiente $r_1(w_1, \sigma_1)$.

- (a.1) Se buscan los valores de α tal que $\alpha_i \neq 0$ y para los que exista un coeficiente $r_h(\alpha, \beta, \Gamma) \neq 0$ verificando $r_j(w_j, \sigma_j) \neq 0 \forall \alpha_j \neq 0$ de α .
- (a.2) Los coeficientes $r_g(a, \beta, c)$ que podrían cambiar debido a la falta son aquellos que, para alguno de los valores de α seleccionados en (a.1), verifican $a_j = \alpha_j w_j$ y $c = \Gamma \prod_{j/\alpha_j=1} \sigma_j$.
- (a.3) Para cada coeficiente $r_g(a, \beta, c)$ seleccionado en (a.2) se comprueba si es impar el número de productos $\prod_i r_{i\alpha_i}(w_i', \sigma_i')$ con $a_i = \alpha_i w_i'$ y $c = \Gamma' \prod \alpha_i \sigma_i'$ que cambian debido a la falta.
- (b) Si una falta de anclaje simple en el módulo H, modifica el coeficiente $r_h(\alpha, \beta, \Gamma)$.**
- (b.1) Se selecciona un conjunto de coeficientes $r_i(w_i, \sigma_i) = 1$ para cada uno de los $\alpha_i = 1$ de α .
- (b.2) Los coeficientes de la función que podrían cambiar son los $r_g(a, \beta, c)$ que verifican $c = \Gamma \prod_{j/\alpha_j=1} \sigma_j$, $a_i = \alpha_i w_i$ si $\alpha_i = 1$, o en caso contrario, $a_i = 0$.
- (b.3) Para cada uno de los r_g seleccionados mediante (b.1) y (b.2) se comprueba que el número de productos $r_h(\alpha', \beta, \Gamma') \prod_i r_i^{\alpha_i'}(w_i', \sigma_i')$, con $\alpha_i w_i = \alpha_i' w_i'$ y $\Gamma' \prod \alpha_i \sigma_i' = \Gamma \prod \alpha_i \sigma_i \forall i$, que cambian es impar. En ese caso, el coeficiente r_g se puede utilizar para detectar la falta.

4.3. Procedimiento modular para la selección de patrones de test.

En la sección 3.3 se describió un procedimiento de selección de los patrones de test para detectar las faltas de anclaje simple en un circuito. Este procedimiento se basa en una heurística de búsqueda de los coeficientes espectrales de la función que cambian frente a las faltas de cada nudo del circuito. Con esos coeficientes se determinaban los patrones de test.

En la presente sección se describe la forma de seleccionar los patrones de test para las faltas de un circuito a partir de los coeficientes espectrales que, en cada módulo, se utilizan para determinar los patrones de test de dicho módulo. Así pues, partiendo de los $r+1$ conjuntos de coeficientes espectrales

$$C(F_i) = \{r_{f_i}(w_i, \sigma_i) / \text{seleccionan los patrones de } F_i\}; \quad (i=1, \dots, r)$$

$$C(H) = \{r_h(\alpha, \beta, \Gamma) / \text{seleccionan los patrones de } H\}$$

el proceso a seguir para obtener el conjunto $C(\emptyset)$, para la función \emptyset sintetizada por todo el circuito, es el siguiente:

Algoritmo 4.2.

1. Para cada $r_h(\alpha, \beta, \Gamma)$ se incluyen en $C(\emptyset)$ todos los coeficientes espectrales $r_\emptyset(a_1, \dots, a_r, b, c)$ cuyos índices verifican

$$\begin{aligned} a_i &= \alpha_i w_i, \\ b &= \beta, \\ c &= \Gamma \cup_i \alpha_i \sigma_i, \end{aligned}$$

para los $r_{f_i}(w_i, \sigma_i) \in C(F_i)$, $i+1, \dots, r$.

2. Para cada coeficiente espectral de $C(F_i)$ y de $C(H)$, marcar el coeficiente espectral de $C(\emptyset)$ con más baja complejidad (de índice con menos unos) de los que lo contengan.

3. Eliminar de $C(\emptyset)$ los coeficientes espectrales no marcados. ■

Los pasos 2 y 3 corresponden a la forma de seleccionar los coeficientes espectrales de manera que el número de patrones de test sea el menor posible y se mantenga en ellos la influencia de todos los coeficientes espectrales de los conjuntos $C(H)$ y $C(F_i)$. La justificación del paso 1 es la siguiente:

(1) Si $r(w_i, \sigma_i)$ está contenido en $C(F_i)$, eso significa que existirá una combinación de entradas al módulo F_i , $(W_i, S_i) \underline{C}(w_i, \sigma_i)$, tal que $f_i(W_i, S_i)$ permite la detección de alguna o algunas de las faltas de anclaje de F_i . Supóngase que $f_i(W_i, S_i) = 0$, cuando se produce la falta la salida será 1. El efecto de ese cambio en la entrada i -ésima de H se puede manifestar en la salida de \emptyset , mediante un patrón de entrada al módulo H , $(A_1, \dots, A_i = 0, \dots, A_r, B, C)$, que detecta la falta de anclaje a 1 en esa entrada. Ese patrón ha de estar seleccionado por alguno de los coeficientes de $C(H)$, $r_h(\alpha, \beta, \Gamma)$, verificando $(A, B, C) \underline{C}(\alpha, \beta, \Gamma)$. Por consiguiente, el patrón que detecte la falta debe estar entre los seleccionados por los coeficientes de \emptyset en los que intervengan $r_h(\alpha, \beta, \Gamma)$ y $r_{f_i}(w_i, \sigma_i)$.

(2) Si $r_h(\alpha, \beta, \Gamma)$ está en $C(H)$ debe haber algún $(A, B, C) \underline{C}(\alpha, \beta, \Gamma)$ que detecte alguna o algunas de las faltas de anclaje de H . Puesto que en cada uno de los conjuntos $C(F_i)$ existen coeficientes espectrales que permiten seleccionar patrones que detecten las faltas de anclaje a 0 y a 1 en las salidas de los módulos F_i , se puede asegurar que si en (A, B, C) , por ejemplo $A_i = 0$, existe un patrón de test $(W_i, S_i) \underline{C}(w_i, \sigma_i)$, para algún $r_{f_i}(w_i, \sigma_i) \in C(F_i)$, que detecta la falta de anclaje a 1 en F_i , y por consiguiente $f_i(W_i, S_i) = 0$. Así pues, el patrón de entradas a \emptyset que detecte la falta se puede encontrar entre los seleccionados por los coeficientes espectrales de \emptyset en los que intervengan $r_h(\alpha, \beta, \Gamma)$ y $r_{f_i}(w_i, \sigma_i)$.

Como ejemplo de aplicación del algoritmo 4.2, considérese el circuito

de la figura 4.3. Los conjuntos de coeficientes espectrales que permiten la detección de las faltas de anclaje simples para cada módulo son los siguientes:

$$C(F1) = \{r_{f1}(10), r_{f1}(01)\},$$

$$C(F2) = \{r_{f2}(100), r_{f2}(010), r_{f2}(101)\},$$

$$C(F3) = \{r_{f3}(11)\}, \text{ y}$$

$$C(H) = \{r_h(101), r_h(011)\}$$

La aplicación del paso 1 del algoritmo 4.2 da como resultado el conjunto $C(\emptyset)$:

$$C(\emptyset) = \{r_\emptyset(1011), r_\emptyset(0111)\}$$

Ambos coeficientes deben seleccionarse puesto que, por ejemplo, $r_{f1}(10)$ no interviene en $r_\emptyset(0111)$, ni $r_{f2}(010)$ en $r_\emptyset(1011)$.

Así pues, el procedimiento permite seleccionar como patrones de test aquellas combinaciones (x, y, z, v) que estén contenidas en (1011) o en (0111) . Se trata de 12 patrones que permiten detectar todas las faltas de anclaje que tienen efecto en la salida del circuito.

4.4. Procedimiento para circuitos con interconexión unidimensional de módulos.

En esta sección se consideran circuitos constituidos por la interconexión unidimensional de módulos, A_1 , tal y como muestra la figura 4.5(a). Las entradas al módulo A_1 se notan como c_{1-1} y a_1 y las salidas corresponden a las funciones $s_1 = s_1(c_{1-1}, a_1)$ y $c_1 = c_1(c_{1-1}, a_1)$, $a_1 \in C_2^{ma}$, $c_1 \in S_2^{mc}$, $s_1 \in C_2^{m1}$. Los coeficientes espectrales de c_1 y s_1 como funciones de las entradas c_{1-1} y a_1 a cada módulo, son conocidos. El problema es el de obtener los coeficientes espectrales en términos de las entradas de todo el circuito c_0, a_1, \dots, a_n . Para este tipo de circuitos también se puede utilizar el tipo de partición propuesta, pero se debe hacer de forma iterativa, tal y como se explica a continuación:

i) Se toman los bloques A_1 y A_2 que definen las funciones $s_2(c_0, a_1, a_2, b)$, $c_2(c_0, a_1, a_2, b)$ y $s_1(c_0, a_1, b)$. En la figura 4.5b el circuito está constituido por un bloque F_1 que corresponde a A_1 y por un bloque H , el A_2 , y los conjuntos de entradas son $X_1 = \{c_0, a_1\}$, $Y = \{a_2\}$ y $Z = b$. De esta forma se pueden utilizar (4.4) y (4.5) para obtener los coeficientes de $s_2(c_0, a_1, a_2, b)$ en términos de $s_2(c_1, a_2, b)$ y $c_1(c_0, a_1, b)$ y los de $c_2(c_0, a_1, a_2, b)$ en términos de los de $c_2(c_1, a_2, b)$ y $c_1(c_0, a_1, b)$.

ii) Los bloques A_1 y A_2 definen el bloque B_2 .

iii) Se repite el proceso para $k=3, \dots, n$ tomando como bloque H el módulo A_k y como bloque F_1 el bloque B_{k-1} construido en el paso anterior, tal y como se muestra en la figura 4.5c. Las variables son $Z=b$, $Y=\{a_k\}$ y $X_1=\{c_0, a_1, \dots, a_{k-1}\}$.

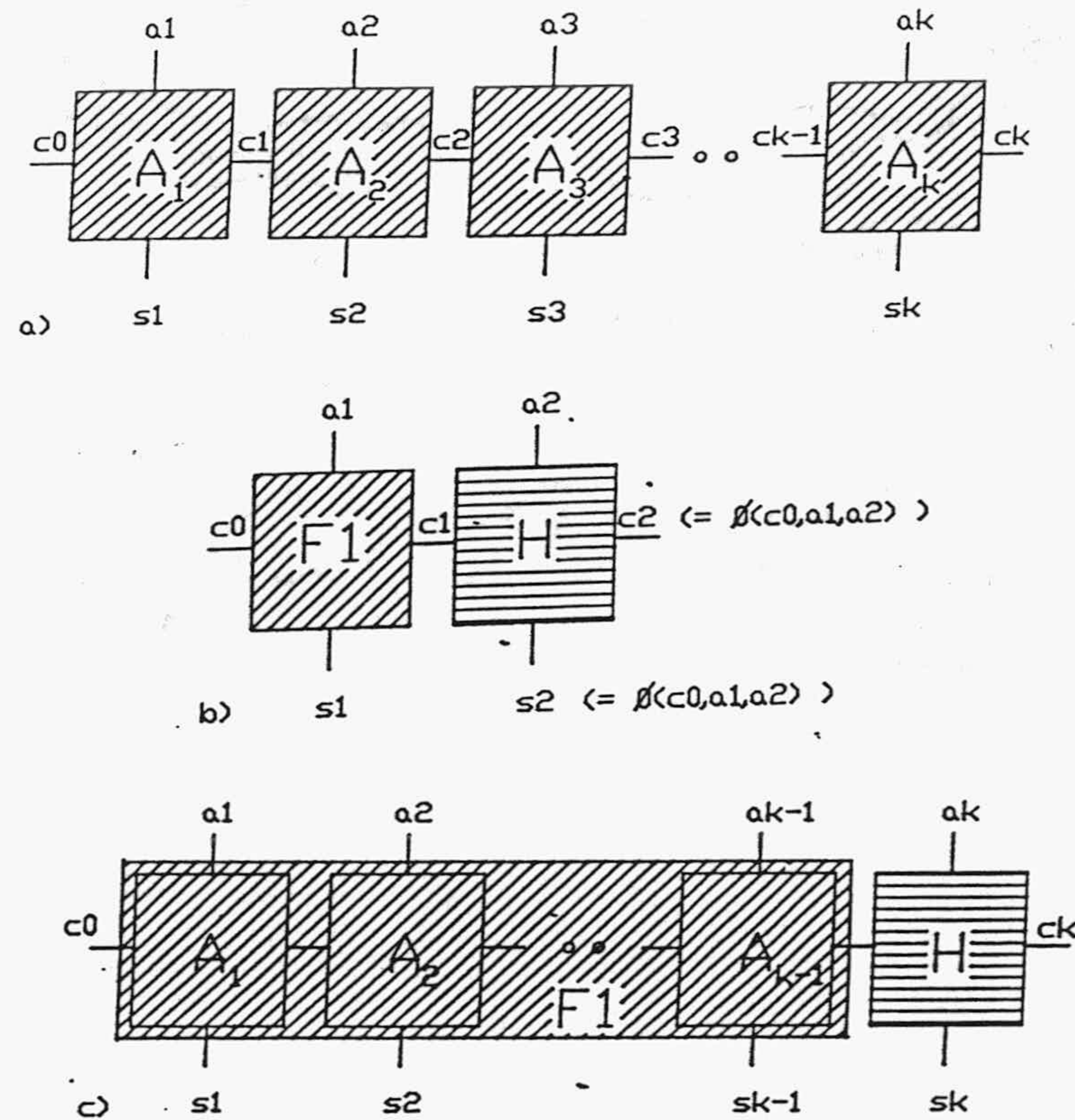


Figura 4.5. Aplicación del procedimiento a un circuito constituido por módulos iguales.

4.5. Ejemplo de aplicación del procedimiento modular.

La figura 4.6a muestra un circuito concreto al que se puede aplicar este proceso. El módulo de circuito que se repite, M_1 en la figura 4.6b, sintetiza las funciones

$$s_1 = s_1(c_{i-1}, a_i, b_1, b_2) = c_{i-1} b_1 \oplus b_1 a_i \oplus b_2 a_i \oplus b_1 b_2 a_i \oplus b_1 b_2 a_i c_{i-1}$$

$$c_1 = c_1(c_{i-1}, a_i, b_1, b_2) = c_{i-1} \oplus a_i \oplus a_{i-1} a_i$$

por lo que los coeficientes espectrales distintos de cero para cada una de las funciones que sintetiza un módulo son:

$$r_s(1010), r_s(0110), r_s(0101), r_s(0111), r_s(1111) \text{ para } s_1,$$

$$\text{y } r_c(1000), r_c(0100), r_c(1100) \text{ para } c_1.$$

Una vez conocidos los coeficientes espectrales de los módulos del circuito, el problema que se plantea consiste en obtener los coeficientes

espectrales del circuito completo a partir de aquellos. El circuito de la figura 4.6(a) se compone de cinco módulos y sintetiza las funciones $(s_1, s_2, s_3, s_4, s_5)$ con ocho entradas $(c_0, a_1, a_2, a_3, a_4, a_5, b_1, b_2)$. Como ejemplo se va a obtener un coeficiente espectral de $s_3(c_0, a_1, a_2, a_3, a_4, a_5, b_1, b_2)$, concretamente el $r_{s_3}(10100010)$.

En primer lugar, se tiene una partición constituida por los módulos M_1 y M_2 que forman el bloque F_1 que sintetiza la función $c_2(c_0, a_1, a_2, b_1, b_2)$ y el módulo M_3 que constituye el bloque H y sintetiza la función $s_3(c_2, a_3, b_1, b_2)$. La agrupación de variables es $X_1 = \{c_0, a_1, a_2\}$, $Y = \{a_3\}$ y $Z = \{b_1, b_2\}$. Las variables a_4 y a_5 no intervienen en s_3 , por consiguiente se las puede considerar dentro del conjunto Y de forma que $Y = \{a_3, a_4, a_5\}$.

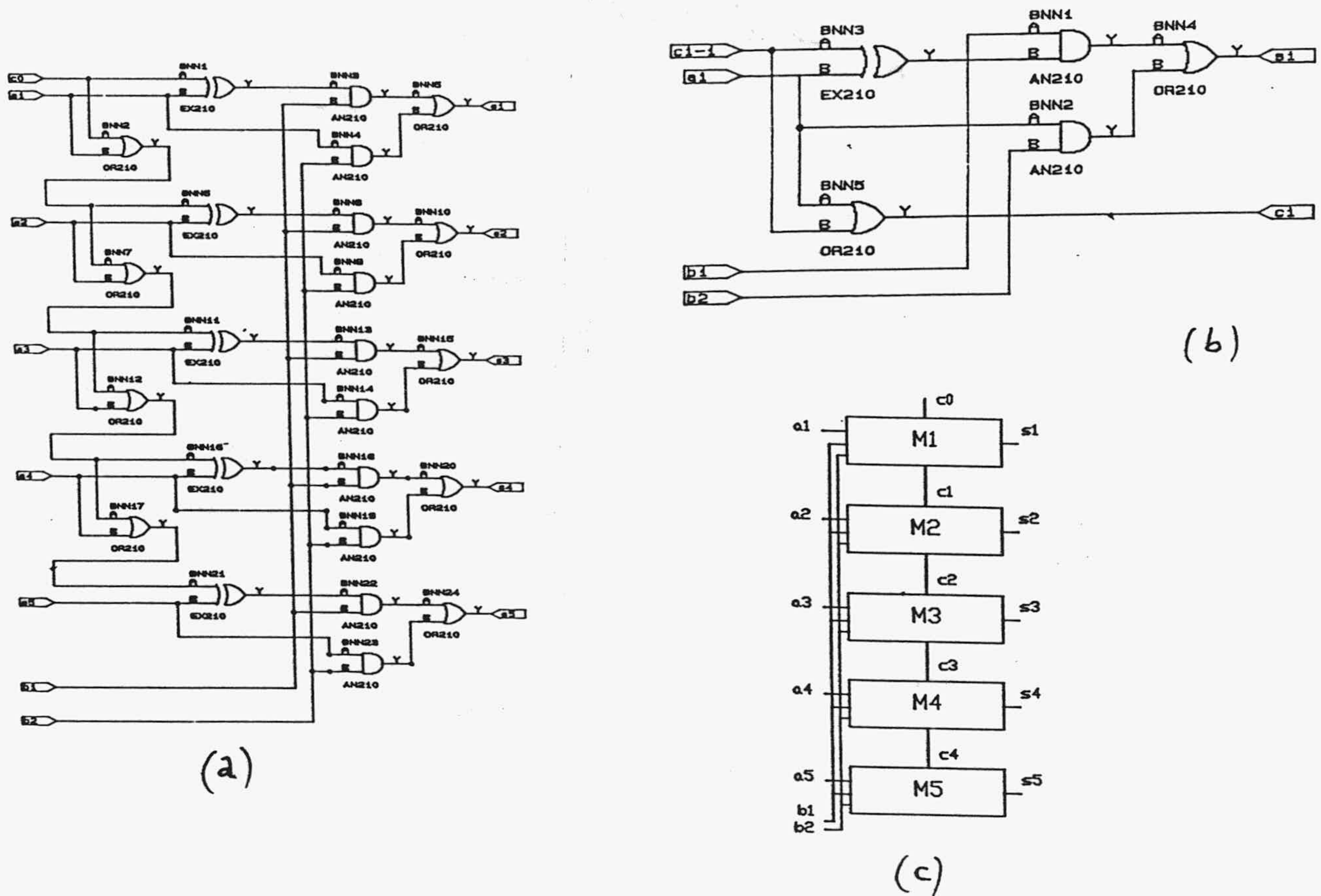


Figura 4.6. Ejemplo de circuito constituido por módulos iguales (a); el módulo constituyente (b); descripción modular (c).

Así, aplicando (4.4) y (4.5) se obtiene que

$$r_{s_3}(10100010) = \sum_{\alpha, \Gamma, w, \sigma} r_s(\alpha, 0, \Gamma_1, \Gamma_2) r_{c_2}(w_0, w_1, w_2, \sigma_1, \sigma_2)$$

con $\alpha w_0 = 1, \alpha w_1 = 0, \alpha w_2 = 1, \Gamma_1 U \alpha \sigma_1 = 1, \Gamma_2 U \alpha \sigma_2 = 0$.

Es decir, $\alpha=1$, $w=(101)$, y los valores posibles para Γ y σ son $\Gamma=(1,0)$, $\sigma=(0,0)$; $\Gamma=(0,0)$, $\sigma=(1,0)$; y $\Gamma=(1,0)$, $\sigma=(1,0)$. Por lo que

$$\begin{aligned} r_{s_3}(10100010) &= r_s(1010)r_{c_2}(10100) \oplus r_s(1000)r_{c_2}(10110) \oplus r_s(1010)r_{c_2}(10110) = \\ &= r_{c_2}(10100) \oplus r_{c_2}(10110) \end{aligned}$$

Para calcular los coeficientes espectrales $r_{c_2}(10100)$ y $r_{c_2}(10110)$ hay que volver a aplicar el proceso con los módulos M_1 y M_2 . El módulo M_1 corresponde al bloque F_1 y el módulo M_2 al bloque H. El bloque F_1 sintetiza la función $c_1(c_0, a_1, b_1, b_2)$ por lo que la agrupación de variables es $X_1=\{c_0, a_1\}$, $Y=\{a_2\}$ y $Z=\{b_1, b_2\}$. Nuevamente se utilizan (4.4) y (4.5) para determinar que

$$\begin{aligned} r_{c_2}(10100) &= r_c(1100)r_c(1000) = 1 \\ r_{c_2}(10110) &= r_c(1110)r_c(1000) \oplus r_c(1110)r_c(1010) \oplus r_c(1010)r_c(1010) = 0 \end{aligned}$$

Por consiguiente $r_{s_3}(10100010)$ es igual a 1. Como se ha podido comprobar a partir del ejemplo, el cálculo de los coeficientes se ha llevado a cabo a partir de las expresiones (4.4) y (4.5) conociéndose únicamente los coeficientes espectrales del módulo o módulos que se utilizan en la síntesis de todo el circuito.

El conjunto de circuitos para los que resulta útil esta forma de cálculo de los coeficientes espectrales es muy amplio, sobre todo teniendo en cuenta que la realización de circuitos complejos se basa, frecuentemente, en la reiteración de módulos más simples. Aunque se han considerado módulos A_1 iguales, el proceso se puede aplicar también con módulos diferentes.

Como ejemplo de aplicación de estas condiciones en el test de las faltas de anclaje utilizaremos el circuito de la figura 4.6. Se sabe que una falta de anclaje a 0 en el nudo BNN1.A modifica los coeficientes $r_s(0010)$, $r_s(1010)$, $r_s(0111)$ y $r_s(1111)$. Se trata de determinar los coeficientes espectrales que se ven afectados por la falta de anclaje a 0 en el nudo 5 del circuito de la figura 4.6(a). Obsérvese que debido a la mayor simplicidad de los módulos constitutivos del circuito es posible obtener información exhaustiva de los coeficientes espectrales que cambian en cada una de las faltas de anclaje. La falta sólo se puede detectar en la salida s_3 , porque el valor del nudo sólo afecta a dicha salida. Por eso sólo hay que buscar los coeficientes espectrales r_{s_3} que cambian.

Para buscar los coeficientes espectrales que cambian se consideran los módulos M_1 y M_2 constituyentes del bloque F_1 , que realiza la función $c_2(c_0, a_1, a_2, b_1, b_2)$, y el módulo M_2 constituyente del bloque H, que realiza

$s_3(c_2, a_3, b_1, b_2)$, y teniendo en cuenta que la falta se produce en el bloque H se aplicará (b).

Si se parte del coeficiente $r_s(0010)$ que cambia debido a la falta considerada, se tiene que $\alpha=0, \beta=000, \Gamma=(10)$. Los coeficientes r_{s_3} que cambian son $r_{s_3}(\alpha w_0, \alpha w_1, \alpha w_2, \beta, 1U\alpha\sigma_1, 0U\alpha\sigma_2)$, es decir sólo el coeficiente $r_{s_3}(00000010)$ podría cambiar. Habría que comprobar que efectivamente es así, para ello se utilizan (4.4) y (4.5) para expresar $r_{s_3}(00000010)$ en términos de los coeficientes r_s y r_c , de esta forma, como:

$$r_{s_3}(00000010) = r_s(0010)$$

Por tanto, $r_{s_3}(00000010)$ permite la detección. Aplicando el mismo procedimiento a $r_s(1010)$, $r_s(0111)$ y $r_s(1111)$ se obtiene que los coeficientes espectrales que permiten la detección de la falta son:

$$r_{s_3}(00000010), r_{s_3}(ddd00010), r_{s_3}(00010011), r_{s_3}(ddd10011)$$

donde las indiferencias se notan con d.

El procedimiento que se ha presentado en esta sección permite relacionar los cambios que se producen en los coeficientes de un módulo de un circuito con los cambios que se producen en los coeficientes de todo el circuito y que se supone que son los accesibles de cara a comprobar la corrección del circuito. Además, puesto que el cálculo de los elementos espectrales se hace sobre partes más simples de un circuito, el número de entradas y el número de nudos es menor y con ello la complejidad del cálculo de los coeficientes.

4.6. Conclusión.

En este capítulo se ha descrito un procedimiento modular para el test de circuitos complejos mediante el empleo de coeficientes espectrales y se ha ilustrado su utilización mediante algunos circuitos de ejemplo.

Si se conocen los coeficientes espectrales de los módulos que constituyen el circuito más complejo, el procedimiento aquí presentado permite obtener los coeficientes del circuito completo. También se ha introducido la heurística para calcular los cambios en los coeficientes espectrales debidos a faltas de anclaje en el circuito partiendo de los cambios que estas faltas ocasionan en los módulos que lo constituyen.

CAPITULO 5. EL TEST CONCURRENTENTE DE CIRCUITOS INTEGRADOS.

La posibilidad de que en cualquier momento de la vida de un circuito integrado se puedan producir defectos, permanentes o temporales [LAL85], y la necesidad de disponer de sistemas electrónicos fiables exigen que los circuitos no sólo se testeen en el momento de su fabricación, sino que conviene que se puedan comprobar periódica o incluso constantemente durante su funcionamiento. Es decir, se hace necesario el **test on-line o concurrente de los circuitos**. Por otra parte, teniendo en cuenta que normalmente los circuitos integrados se insertan dentro de tarjetas en el interior de sistemas, resulta impracticable extraer el circuito para conectarlo a un sistema de aplicación de test cada vez que se va a testear. En ese sentido, las técnicas de test BIST off-line de circuitos integrados [MCL85a, 85b], además de facilitar el acceso a los módulos del circuito integrado, posibilitan el test periódico. No obstante, los defectos temporales, y más concretamente los intermitentes, no se pueden detectar por esta técnica sino que precisan de un test concurrente. Teniendo en cuenta que a medida que los circuitos integrados se hacen más pequeños estos defectos son más frecuentes [ABR86b], aumenta la necesidad de desarrollar técnicas eficientes de test concurrente [FUC87, FUJ84, 87, KHA82, SAL88, SAY86, SHI84].

La capacidad de funcionar de manera fiable aún en presencia de fallos es esencial en sistemas que deban operar en ambientes ruidosos o que se quieran muy seguros y que, en el caso extremo, dejen de funcionar sin efectos perniciosos para su entorno [BEC89, BLO90, LAL85, SHI84]. Estos sistemas

reciben el nombre de **sistemas tolerantes a fallos** [AVI82, BER87]. Además, el desarrollo de la tecnología WSI (Wafer Scale Integration) ha hecho imprescindible la tolerancia a fallos de todo el sistema incluido en una sola oblea como una forma de aumentar el rendimiento del proceso de fabricación [KOR86]. Los sistemas tolerantes a fallos, mientras estén funcionando normalmente, deben tener la capacidad de [SHI84]:

- (a) Detectar la presencia de defectos,
- (b) Localizar esos defectos (realizar la diagnosis del defecto),
- (c) Reconfigurarse de manera que se aisle la parte defectuosa, y
- (d) Recuperarse del error, es decir, anular los efectos del defecto antes de que se detectase.

Así pues, la posibilidad de detectar un error en el momento en que se produce o lo antes posible, constituye el punto de partida para la tolerancia a los fallos. El principio básico de los mecanismos de detección que utilizan estos sistemas es la **redundancia** [SHI84], que puede ser de **dispositivos, información o de tiempo**, y se pueden dividir en:

(a) Mecanismos de detección a nivel de señal: que implican la existencia de una circuitería adicional que permita el test concurrente de los módulos del sistema. Entre los métodos de test concurrente se encuentran, por ejemplo, los basados en códigos detectores/correctores de errores o en la duplicación del circuito, y su efectividad se mide a través de la probabilidad de detectar un error inducido por un defecto arbitrario del circuito.

(b) Mecanismos de detección a nivel funcional: en los que la detección se realiza a un nivel superior, normalmente mediante una circuitería específica y atendiendo a aspectos de funcionamiento del sistema, como **tiempo de respuesta, resultados de operación probables, etc.**

(c) Mecanismos de comprobación periódica: que se basan en procedimientos software consistentes en la ejecución periódica de ciertos programas de diagnosis o en procedimientos de test off-line de los circuitos del sistema que se aplican también de forma periódica. Interesa, de cara a la aplicabilidad del test, que los procedimientos de test sean BIST.

El test concurrente de los módulos que constituyen el sistema permite que se puedan detectar los errores mientras el sistema está funcionando, e incluso ciertos procedimientos de test concurrente permiten llevar a cabo la corrección on-line de estos errores. Los métodos de test concurrente necesitan de circuitos chequeantes autocomprobados [BOS84d, HUG84, JHA89, PIE87, TAM84], con un elevado cubrimiento de faltas y realizables con un hardware mínimo.

Son muchas las publicaciones en el área de la detección concurrente de

errores, habiéndose concentrado la mayoría de ellas en el diseño de circuitos chequeantes para diferentes códigos y para unidades funcionales simples con entradas y salidas codificadas [AND71, MAR78, WAN79, WON83, KHA84, SAY86, ELL90]. En cambio, hay menos trabajos en que presenten métodos de test concurrente para circuitos arbitrarios, con entradas y salidas sin codificar. Entre estos se encuentran un método general para la predicción de la paridad de la salida de un circuito combinacional [FUJ83], una técnica de comprobación de paridad extendida para la detección de un número impar de errores [K078] y un circuito chequeante autocomprobado de predicción generalizada (GPC) que permite la detección de errores múltiples [FUJ87]; en otro trabajo [SAL88] se aprovecha la misma circuitería utilizada para el test BIST off-line con algunos módulos adicionales para realizar el test concurrente.

En este capítulo se presentan los conceptos y definiciones utilizados en el contexto del test concurrente de circuitos y su relación con la tolerancia a fallos. Se presenta un modelo del proceso de detección que incluye la posibilidad de defectos permanentes y temporales (transitorios e intermitentes) y a partir de él se obtendrán medidas para la evaluación de los procedimientos de test concurrente. Se describen las características generales de los procedimientos que hasta el momento se han propuesto y se presenta, comparándose con los anteriores, el que se ha elaborado en la presente memoria. En los capítulos 6 y 7 se muestra la forma de realizar el diseño de los módulos a añadir a un circuito para hacerlo testable concurrentemente. A través de estos tres capítulos (5, 6 y 7) también se pretende mostrar la relación entre los procedimientos de compresión de las salidas y de generación de patrones de test en el diseño de circuitos testables concurrentemente.

5.1. Un modelo del proceso de detección.

Un sistema tolerante a fallos que pueda presentar defectos permanentes y temporales se puede encontrar en alguno de los siguientes estados:

- (a) No presenta ningún defecto (ND).
- (b) Presenta un defecto latente (DL).
- (c) Presenta un defecto temporal intermitente no activo y no se ha producido error (DI).
- (d) Se ha producido una salida errónea no detectada, es decir, existe un error latente (E).
- (e) Se ha producido una salida errónea detectada (D1).

donde intervienen las siguientes probabilidades:

- α_1 , se produzca un defecto latente en el circuito;
- α_2 , un defecto transitorio desaparezca;
- β_1 , un defecto intermitente se haga activo;
- β_2 , un defecto intermitente pase a no ser activo;
- σ , se produzca un defecto intermitente no activo;
- μ_1 , un defecto intermitente se haga activo existiendo un error latente;
- μ_2 , un defecto intermitente deje de ser activo existiendo un error latente;
- p_1 , al aplicar una entrada al circuito se genere un error no detectado;
- p_2 , al aplicar una entrada al circuito se genere un error y se detecte;
- p_3 , al aplicar una entrada al circuito se genere un error no detectado habiéndose producido con anterioridad otros errores no detectados.

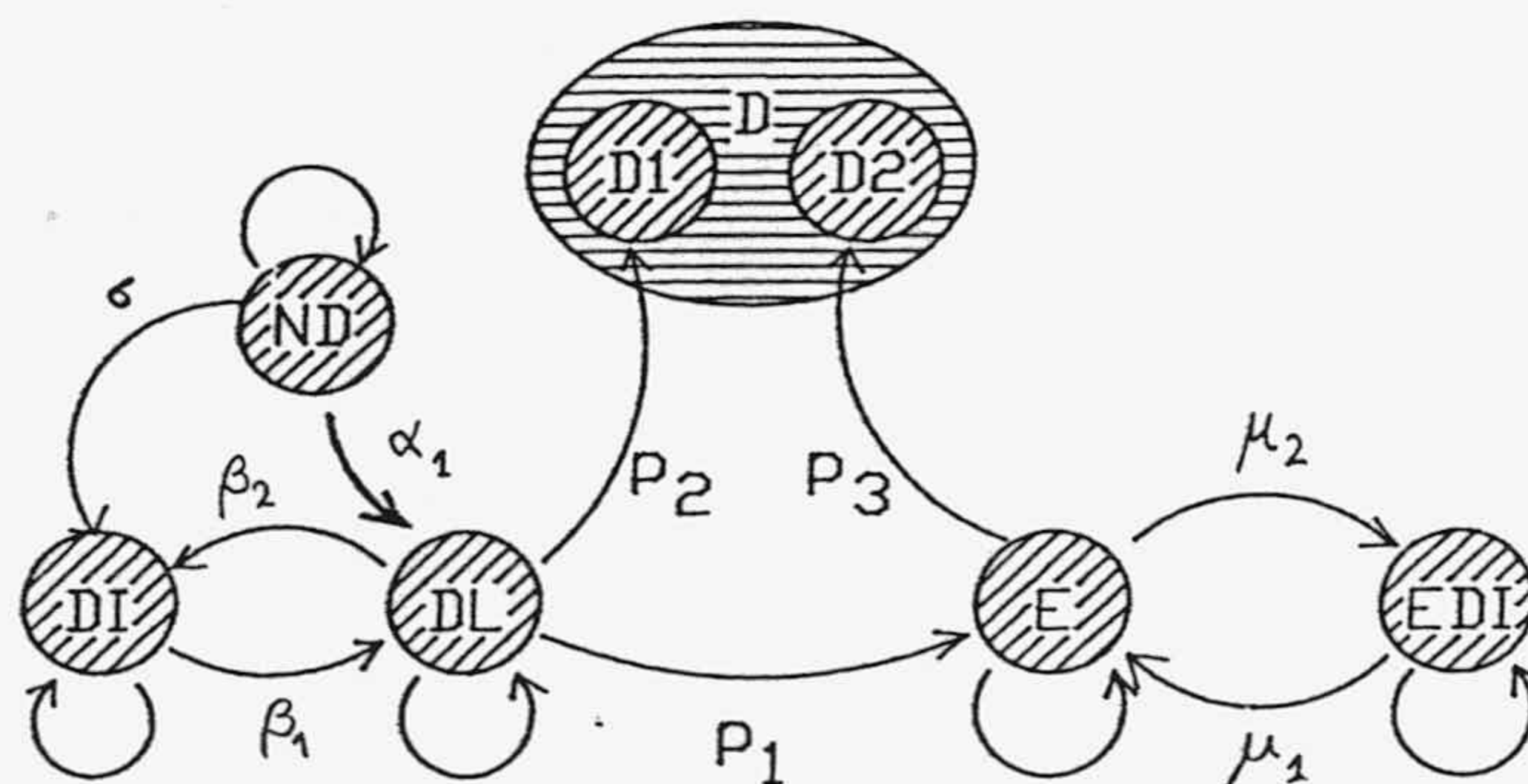


Figura 5.2. Diagrama cuando no hay faltas transitorias.

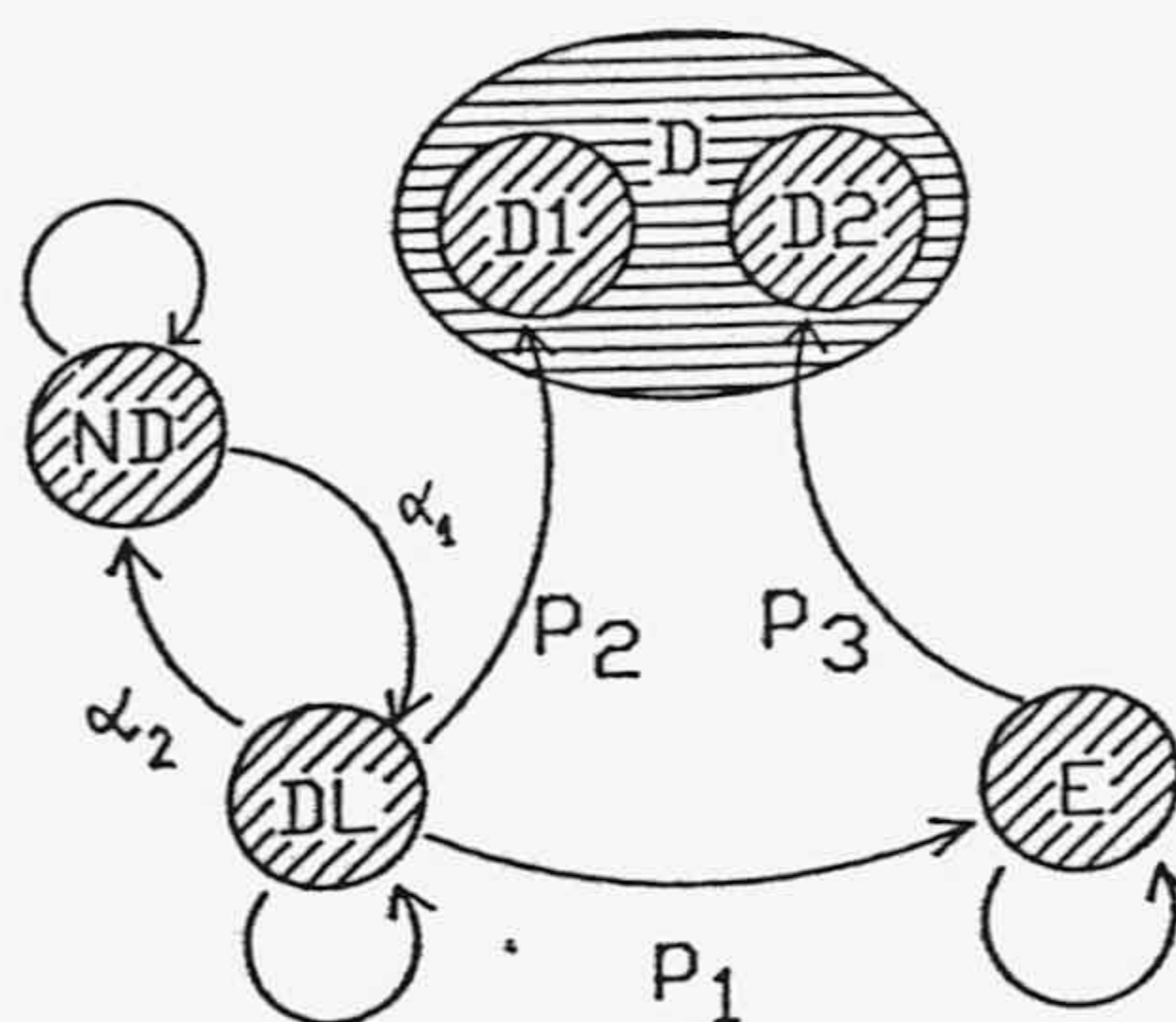


Figura 5.3. Diagrama cuando no hay faltas intermitentes.

El diagrama de la figura 5.1 se simplifica en ciertos casos. Por ejemplo, si no existen faltas temporales transitorias, no puede haber transición entre el estado DL y ND (figura 5.2), y si se considera que no existe la posibilidad de faltas intermitentes se pueden eliminar los estados DI y EDI (figura 5.3). Por último, si sólo interesa evaluar la detección de defectos, se pueden agrupar los estados ND y DI en un único estado, N, y los estados E y EDI en otro único estado, E (figura 5.4).

Dado el vector de probabilidades en el instante i -ésimo

$$\mathbf{n}(i) = (n_1(i), n_2(i), n_3(i), n_4(i), n_5(i), n_6(i), n_7(i)),$$

donde $\pi_j(i)$ es la probabilidad de estar en los siguientes estados:
 $\pi_1(i)$, ND; $\pi_2(i)$, DL; $\pi_3(i)$, DI; $\pi_4(i)$, E; $\pi_5(i)$, EDI; $\pi_6(i)$, D1; y $\pi_7(i)$, D2;
 de forma que a partir de la matriz de probabilidades P y del vector de probabilidades en el instante i, se puede obtener el vector de probabilidades en el instante i+1 según la expresión:

$$\pi(i+1) = \pi(i) \cdot P \tag{5.1}$$

De esta forma se tiene que

$$\delta\pi(i) = \pi(i+1) - \pi(i) = \pi(i) \cdot P - \pi(i) = \pi(i) \cdot (P - I) \tag{5.2}$$

y tomando $\delta t = 1$,

$$\delta\pi(i) / \delta t = \pi(i) \cdot M \tag{5.3}$$

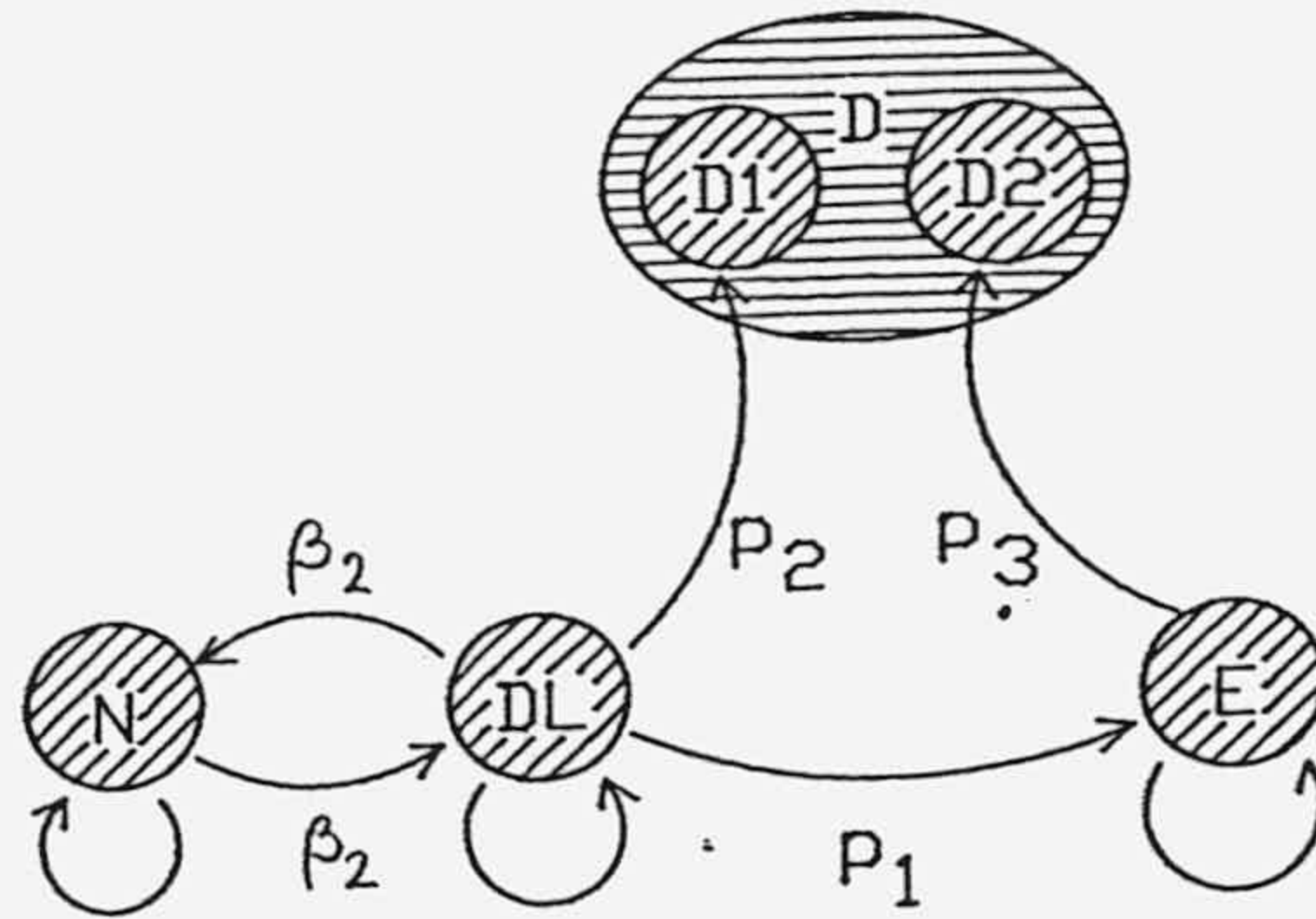


Figura 5.4. Diagrama para la detección de defectos.

En (5.3) la matriz M recibe el nombre de **matriz de transición de probabilidades**, y como se puede ver a partir de (5.2), se obtiene restando a la matriz de probabilidades, P, la matriz identidad, I:

$$\begin{bmatrix} -(\sigma + \alpha_1) & \alpha_1 & \sigma & 0 & 0 & 0 & 0 \\ \alpha_2 & -(\alpha_2 + \beta_2 + p_1 + p_2) & \beta_2 & p_1 & 0 & p_2 & 0 \\ 0 & \beta_1 & -\beta_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -(\mu_2 + p_3) & \mu_2 & 0 & p_3 \\ 0 & 0 & 0 & \mu_1 & -\mu_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Resolviendo el sistema de ecuaciones diferenciales (5.3) se puede obtener la variación con el tiempo de los valores de las probabilidades de los distintos estados y, particularmente, de los estados de detección, D1 ó D2. Considerando lo rápido que se consiga un valor aceptable para la probabilidad de detección se puede evaluar el procedimiento de detección utilizado.

Se puede reducir en uno el número de ecuaciones si se consideran los estados ND y DI como un sólo estado, que seguiremos llamando ND y que

corresponde a la situación en la que o bien no existan defectos o, si los hubiera, no se encuentren activos. Con esto, la nueva matriz M es:

$$\begin{bmatrix} -(\alpha_1 + \beta_1) & \alpha_1 + \beta_1 & 0 & 0 & 0 & 0 & 0 \\ \alpha_2 + \beta_2 & -(\alpha_2 + \beta_2 + p_1 + p_2) & 0 & p_1 & 0 & p_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -(\mu_2 + p_3) & \mu_2 & 0 & p_3 \\ 0 & 0 & 0 & \mu_1 & -\mu_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

El sistema de ecuaciones a resolver es:

$$\begin{aligned} \frac{d\pi_1}{dt} &= -a\pi_1 + b\pi_2 \\ \frac{d\pi_2}{dt} &= a\pi_1 - (b + p_1 + p_2)\pi_2 \\ \frac{d\pi_4}{dt} &= p_1\pi_2 - (\mu_2 + p_3)\pi_4 + \mu_1\pi_5 \\ \frac{d\pi_5}{dt} &= \mu_2\pi_4 - \mu_1\pi_5 \\ \frac{d\pi_6}{dt} &= p_2\pi_2 \\ \frac{d\pi_7}{dt} &= p_3\pi_4 \end{aligned} \quad (a = \alpha_1 + \beta_1 \quad b = \alpha_2 + \beta_2) \quad (5.4)$$

Para resolverlo, se utiliza la transformada de Laplace teniendo en cuenta que inicialmente las probabilidades de los estados 4, 5, 6 y 7 son nulas ya que, al menos hasta que no se aplique alguna combinación de entradas no se puede producir ningún error ($\pi_4(0) = \pi_5(0) = \pi_6(0) = \pi_7(0) = 0$). Como se está suponiendo que las probabilidades de transición son independientes del tiempo, las expresiones para las transformadas de Laplace de las soluciones $\pi_6(t)$ y $\pi_7(t)$ son las siguientes:

$$\begin{aligned} \pi_6(s) &= (p_2/s) [(s\pi_2(0) + a) / (s^2 + (a + b + p_1 + p_2)s + a(p_1 + p_2))] \\ \pi_7(s) &= (p_3/s) [(s + \mu_1)p_1 / (s^2 + (\mu_1 + \mu_2 + p_3)s + \mu_1 p_3)] \\ &\quad [(s\pi_2(0) + a) / (s^2 + (a + b + p_1 + p_2)s + a(p_1 + p_2))] \end{aligned} \quad (5.5)$$

Calculando la transformada inversa, se pueden obtener la función de variación de las probabilidades con el tiempo. A partir de $\pi_6(t)$ se puede determinar el tiempo que tardaría en detectarse un error con una probabilidad dada, sin que se haya generado ningún error previo no detectado, y a partir de $\pi_7(t)$ se puede obtener la misma información, pero en el caso de que previamente se haya producido un error no detectado. La probabilidad de detección de error en un momento dado, se obtiene a partir de $\pi_6(t) + \pi_7(t)$. La forma de las soluciones es

$$\begin{aligned} \pi_6(t) &= A + (Be^{w_1 t}) + (Ce^{w_2 t}) \\ \pi_7(t) &= D + (Ee^{w_1 t}) + (Fe^{w_2 t}) + (Ge^{\tau_1 t}) + (He^{\tau_2 t}) \end{aligned} \quad (5.6)$$

donde w_1, w_2, τ_1, τ_2 se obtienen a partir de las expresiones de las raíces de los denominadores:

$$\begin{aligned}
w_1 &= -[(a+b)+(p_1+p_2)]/2 + \{(a+b+p_1+p_2)^2 - 4a(p_1+p_2)\}^{1/2}/2 \\
w_2 &= -[(a+b)+(p_1+p_2)]/2 - \{(a+b+p_1+p_2)^2 - 4a(p_1+p_2)\}^{1/2}/2 \\
\tau_1 &= -[(\mu_1+\mu_2)+p_3]/2 + \{(\mu_1+\mu_2+p_3)^2 - 4\mu_1 p_3\}^{1/2}/2 \\
\tau_2 &= -[(\mu_1+\mu_2)+p_3]/2 - \{(\mu_1+\mu_2+p_3)^2 - 4\mu_1 p_3\}^{1/2}/2
\end{aligned} \tag{5.7}$$

Los valores de w_1 , w_2 , τ_1 y τ_2 son siempre reales y negativos, por lo que los términos exponenciales serán despreciables a partir de cierto instante, que será tanto menor cuanto mayores sean los w y los τ . Los valores de las raíces dependen, por una parte, de p_1 , p_2 y p_3 , que son función de las características del procedimiento de detección y, por otra, de μ_1 , μ_2 , a y b , que están ligados a las características físicas de los circuitos y al entorno en el que funcionan. Por consiguiente, son los valores de las probabilidades p_1 , p_2 , y p_3 los que se pueden controlar, desde el punto de vista del diseño del test, para disminuir el tiempo que tarda en obtenerse un valor suficientemente alto en la probabilidad de detección. Teniendo en cuenta que los valores absolutos de w_1 y w_2 aumentan con p_2 y los valores de τ_1 y τ_2 aumentan con p_3 , interesa aumentar tanto una probabilidad de transición como la otra para mejorar la capacidad de detección del sistema. Los valores de las constantes se obtienen realizando la transformada de Laplace inversa de $\pi_6(s)$ y $\pi_7(s)$ y han de verificar además que

$$\pi_6(0)=0=A+B+C; \quad \pi_7(0)=0=D+E+F+G+H; \quad \pi_6(\infty)+\pi_7(\infty)=1=A+D$$

Hasta ahora, se ha supuesto que las probabilidades de transición en la matriz M son constantes. Esto es totalmente válido en el caso de que se consideren defectos permanentes, se supone que se ha producido un defecto concreto y la capacidad de ocasionar un error en las salidas es la misma para todas las combinaciones de entradas que se aplican. Para los demás casos hay que considerar promedios referidos a todos los posibles defectos que se puedan producir y a la capacidad de detección de las entradas. En cualquier caso, el interés de este desarrollo se centra en relacionar la capacidad de detección del procedimiento de test con las probabilidades de detección p_2 y p_3 . Estas probabilidades se refieren a la capacidad de que una entrada aplicada al circuito genere un error detectable por el sistema de test concurrente y dependen de la detectabilidad del defecto que presente el circuito y de la capacidad del módulo de reconocimiento de errores.

Si el sistema de reconocimiento de errores sólo actúa sobre las salidas que se obtienen en cada momento, la probabilidad de detectar un error es independiente de que se haya producido un error anteriormente. Por consiguiente, se tendrá que $p_2=p_3=pd$, siendo pd la probabilidad de detección

de error por parte de una entrada, que se puede expresar a partir de la probabilidad de que una entrada detecte un defecto (dé lugar a un error), p_e , y de la probabilidad de enmascaramiento del sistema de análisis de las respuestas (sección 1.4), p_A :

$$p_d = p_e (1 - p_A) \quad (5.8)$$

De esta forma, puesto que la probabilidad p_1 de la matriz M corresponde a la probabilidad de que una entrada genere un error que no se detecte, se tiene que:

$$p_1 = p_e p_A \quad (5.9)$$

5.2. Los circuitos testeables concurrentemente.

Los circuitos lógicos que poseen la capacidad de detección de faltas durante su funcionamiento se denominan [LAL85] circuitos **autocomprobantes (concurrentemente)**. Los procedimientos de test previamente propuestos en la literatura se pueden encuadrar dentro de los dos tipos siguientes:

5.2.1. Procedimientos Tipo I.

Por una parte están los procedimientos, que se denominarán **Tipo I**, en los que el circuito se diseña para que sus salidas pertenezcan a un código elegido de manera que las erróneas no pertenezcan a ese código. Posteriormente, un circuito detecta si la salida pertenece al código (no hay defecto o el defecto no ha ocasionado error) o no (sí hay defecto). Entre los códigos que se utilizan en estos procedimientos de test concurrente están los códigos **Hamming** de detección y corrección de errores **SEC/DED** [ELL90], códigos **resíduos** para circuitos aritméticos, ALUs, etc. [SAY86], códigos de **Berger** para la detección de errores unidireccionales (sólo se producen errores en los que los ceros se han transformado en unos o los unos en cero) [FUC87] utilizados en PLAs, códigos de **Borden** para detectar errores unidireccionales con un máximo número de bits erróneos [JHA89]. Las características de estos códigos se pueden consultar, además de en los trabajos referenciados, en textos específicos como [LIN83, PET72]. Además se pueden determinar los códigos óptimos para detectar y corregir errores unidireccionales [BOS82c, PRA80], que se consideran los patrones de error más probables en circuitos VLSI [JHA89]. Existe una infinidad de trabajos sobre códigos utilizables en

la detección de distintos tipos de patrones de errores, entre los que están [NAN87, NIK88, PIE87, TAO88, BOR87, YIN88].

No obstante en circuitos complejos, como es el caso de un microprocesador, los patrones de error que se observan no se ajustan a un único tipo y así, los distintos módulos darán lugar a diferentes comportamientos erróneos [TAM84]. Además, la mayoría de los trabajos presentados no consideran de forma realista la complejidad del circuito de comprobación de código y la debida a la necesidad de codificar las salidas del circuito. En un trabajo reciente [ELL90] la circuitería añadida para testear concurrentemente un RISC mediante el uso de códigos Hamming SEC/DED supone un 112% del circuito a testear, es decir corresponde a más de la mitad del chip. En otro trabajo [SAY86] se indica que el incremento de circuitería puede llegar a ser un 70% del circuito a testear. Parece ser, por consiguiente, que esta forma de diseñar los circuitos testables concurrentemente puede dar lugar a circuitos bastante más complejos que el original.

En la figura 5.5 se muestra un circuito a testear (CUT) y los cambios que se precisan para que sea testeable con un procedimiento Tipo I. La salida se construye con la salida X del circuito inicial y con una serie de salidas adicionales δX tales que $X' = (X, \delta X)$ ha de ser una palabra de un código C , para cualquier salida correcta del circuito. El circuito CC de la figura es el que comprueba si la salida $(X, \delta X)$ pertenece al código C ; en caso de que la salida X' no pertenezca al código C generará una señal de error.

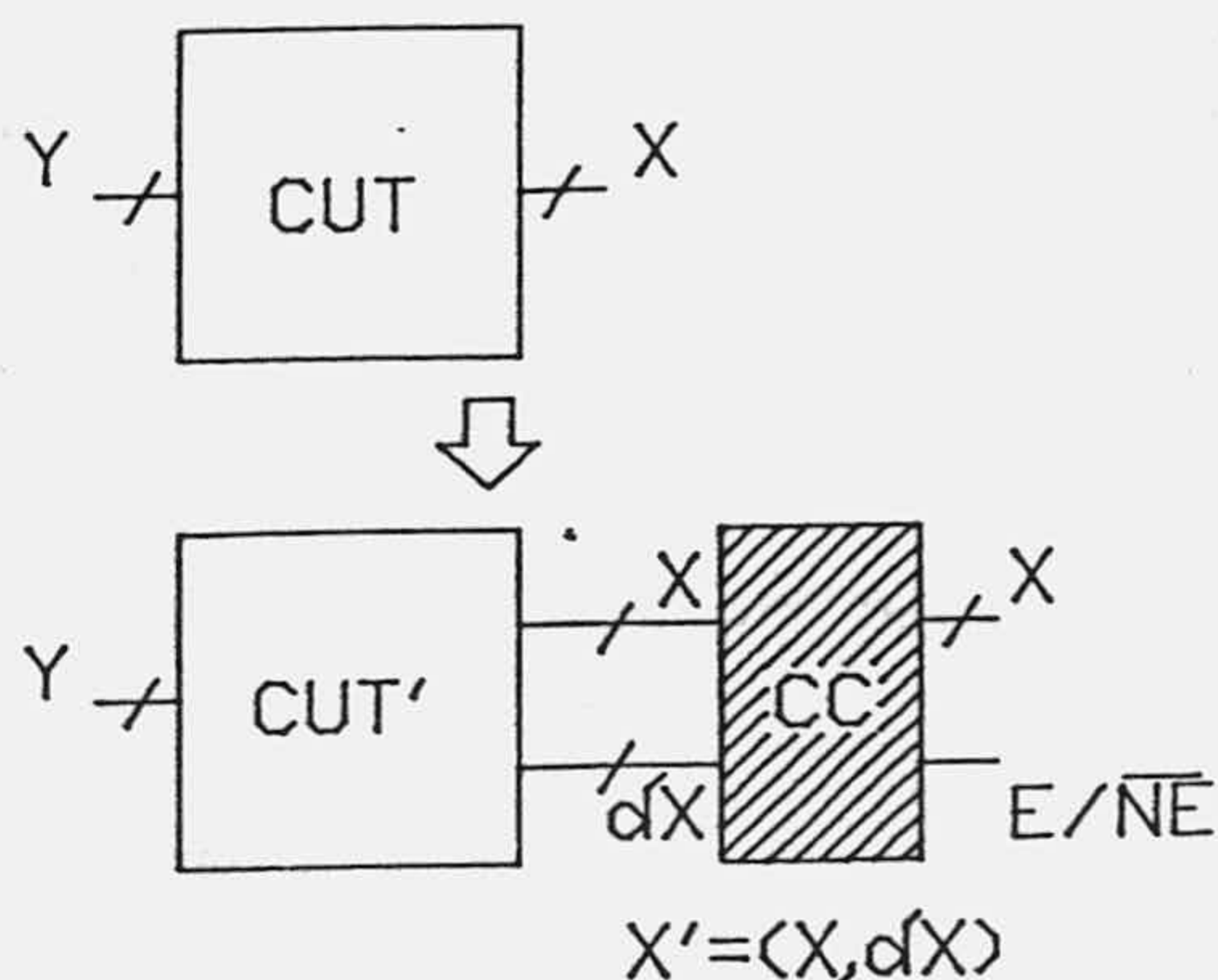


Figura 5.5. Modificación de un circuito para su test tipo I.

Según las propiedades de detección que exhiba un sistema de test concurrente, se puede calificar con alguno o varios de los siguientes términos, que constituyen definiciones más precisas y formales de la capacidad de detección de defectos concurrente:

(a) **Circuito autocomprobante para un conjunto F de faltas** es aquel circuito que, durante su funcionamiento, produce alguna salida que no pertenece al código elegido si se ve afectado por una cualquiera de las faltas del conjunto F. Es decir, $\forall f \in F$ existe una salida $X^* = (X^*, \delta X^*) / X^*$ no pertenece a C.

(b) **Circuito seguro frente a un conjunto F de faltas**, es aquel circuito que nunca produce una salida incorrecta que sea palabra del código si se ve afectado por una cualquiera de las faltas del conjunto F: $\forall f \in F$ si $CUT(Y) \neq CUT^*(Y)$ entonces $CUT^*(Y)$ no pertenece a C.

(c) **Circuito totalmente autocomprobante para un conjunto F de faltas** es un circuito autocomprobante que, además es seguro frente a las faltas del conjunto F. Estos circuitos se designarán circuitos TSC, atendiendo a sus iniciales en inglés (Totally Self-Checking). Un circuito seguro frente a F, es TSC si para alguna entrada Y se verifica que $CUT(Y) \neq CUT^*(Y)$.

(d) **Circuito muy seguro frente a las faltas de un conjunto, F**, dado es aquel que, para cualquier falta, f, del conjunto F verifica:

d1) es TSC para el conjunto de faltas {f}, o

d2) es seguro frente a faltas para {f} y si el circuito se ve afectado por f el circuito sigue siendo muy seguro frente a faltas del conjunto $F - \{f\}$.

Los circuitos muy seguros frente a faltas se designarán como SFS, del inglés Strongly Fault Secure. Un circuito es SFS si es seguro frente a una cierta serie de faltas hasta que se produce una falta que haga que el circuito sea autocomprobante para el resto de faltas.

5.2.2. Procedimientos Tipo II.

Otra forma de realizar el test concurrente se presenta en [SAL88]: son los circuitos testeables concurrentemente mediante **procedimientos Tipo II**. Como se puede ver en la figura 5.6, se utiliza un LFSR para comprimir las salidas del circuito y otro para indicar qué entrada de test debe ser la siguiente a aplicar. Cuando la entrada al circuito coincide con el estado del LFSR que indica el patrón de test siguiente, una señal activa el LFSR que analiza las salidas del circuito y el LFSR de la entrada cambia de estado indicando el siguiente patrón. Una vez aplicados todos los patrones que constituyen el test se comprueba si la signatura almacenada en el LFSR de salida corresponde a la del circuito correcto. Este procedimiento aprovecha la misma circuitería del test BIST off-line para realizar el test concurrente por lo que la complejidad de hardware adicional no es elevada. El problema es

que el test sólo se completa después de ciertos intervalos de tiempo y además deben aplicarse los patrones según un orden establecido, lo que implica que en cada instante, de todas las combinaciones de entrada posibles, sólo una hace progresar el test. Si el número de entradas del circuito es muy elevado, y todas las combinaciones son equiprobables, la probabilidad de que se aplique una entrada que es patrón de test es muy pequeña.

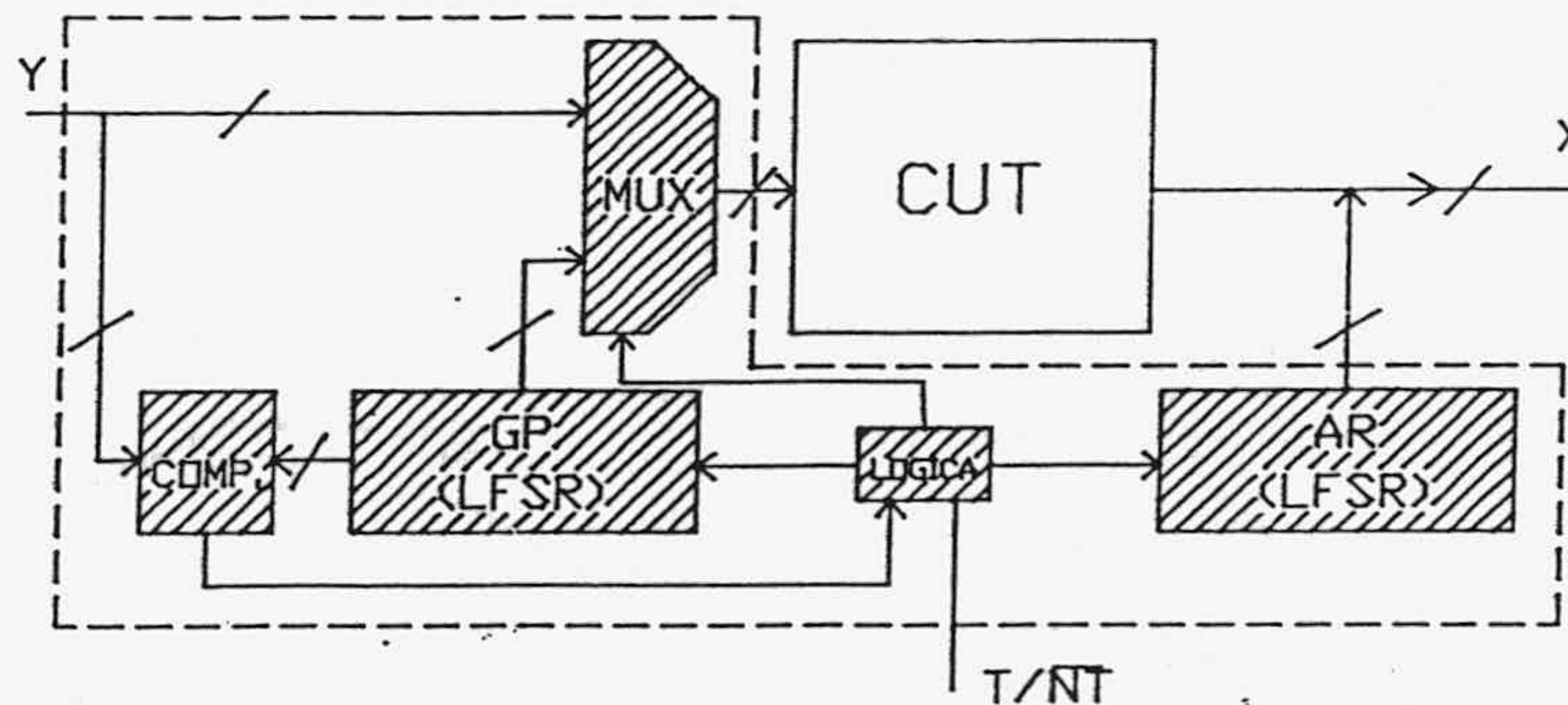


Figura 5.6. Circuitería añadida a un circuito para el test tipo II.

5.3. Evaluación de los procedimientos de Test Concurrente.

En esta sección se define una serie de parámetros que permiten evaluar un procedimiento de test concurrente. Gracias a ellos se pueden comparar los distintos procedimientos de test concurrente que se pueden aplicar a un circuito dado y seleccionar el más adecuado. Se trata de:

(A) El tiempo de latencia (TL), tomado como el número medio de entradas que habría que aplicar desde el momento en que se produce un defecto en el circuito hasta que dicho defecto se detecta. Obviamente, cuanto menor sea el tiempo de latencia mejor es el procedimiento de test.

(B) Incremento de complejidad del circuito (δC), referido al incremento relativo de superficie de silicio que se necesita para implementar físicamente el procedimiento de test concurrente. A veces, esta medida se obtiene de forma indirecta evaluando el incremento en el número de puertas lógicas, el de líneas de producto en el caso de PLAs, etc.

(C) Disminución de prestaciones (δD), medido como el incremento relativo del retardo que presenta el circuito que se testea.

(D) Tiempo de desarrollo (T_d), o tiempo necesario para obtener la descripción del circuito testeable concurrentemente más el incremento de tiempo en el

iseño físico del mismo.

De estos cuatro parametros, en la presente memoria se consideran los dos primeros (TL y δC), fundamentalmente, debido a que son los que se tienen en cuenta de forma más directa como parámetros de diseño. La disminución en las prestaciones se considera de manera implícita al buscar una complejidad mínima en la circuitería añadida. Además, se pueden introducir ciertas modificaciones que eviten excesivos retardos una vez se ha determinado la forma de los módulos de predicción, extracción y monitorización de patrones; por ejemplo, establecer una segmentación (pipelinig) entre la evaluación de la función y la comprobación de las salidas. En cuanto al tiempo de desarrollo, este se puede evaluar a partir de la complejidad de los algoritmos que se utilizan en el diseño del circuito testeable concurrentemente.

5.3.1. Estimación del tiempo de latencia (TL).

En este apartado se considera la evaluación de TL en un procedimiento de test concurrente cuando el circuito presenta defectos permanentes.

El tiempo de latencia es el número medio de entradas aplicadas entre que se ha producido un defecto y se aplica una de las entradas que lo detecta. Si pd es la probabilidad de que se detecte un defecto al aplicar una entrada, la probabilidad de detectar ese fallo tras aplicar exactamente q entradas, $PD(q)$, vendrá dada por:

$$PD(q) = (1-pd)^{q-1} pd \quad (5.10)$$

Con ello, el número medio de entradas para detectar un defecto desde el momento en que éste se produjo es:

$$TL = \sum_{q=1}^{\infty} q PD(q) = \sum_{q=1}^{\infty} q (1-pd)^{q-1} pd = 1/pd \quad (5.11)$$

Para completar la evaluación de TL, es preciso expresar pd de manera que se refleje la capacidad de los patrones de test para generar salidas erróneas en el caso de que se presente algún defecto y la capacidad de detección de esas salidas erróneas por parte de los circuitos extractor y predictor añadidos. Para ello, se utiliza (5.8):

$$pd = p_e ps \quad (5.12)$$

donde p_e es la probabilidad de que la entrada aplicada sea de test que dé



lugar a un error, y p_s la probabilidad de que exista diferencia entre las salidas del circuito de extracción y las del de predicción. Para obtener p_e se puede utilizar la expresión siguiente:

$$p_e = \sum_{f \in F} p(f) n_f(T) / L \quad (5.13)$$

En (5.13), la suma se extiende a todos los defectos modelados como faltas, f , del conjunto que se considera modela los defectos del circuito, L es el número de patrones del conjunto de test, T , $p(f)$ es la probabilidad de que se produzca la falta f y $n_f(T)$ es el número de patrones del conjunto de test T que dan lugar a un error en las salidas del circuito cuando se ve afectado por la falta f . El problema que plantea la evaluación de (5.13) es la determinación de los valores de $n_f(T)$ para el test T y para todos los posibles defectos del circuito consideradas. La solución adoptada para resolver este problema se describe en el capítulo 7 de la presente memoria.

El cálculo de p_s también presenta ciertos problemas. La expresión a utilizar para evaluarlo es:

$$p_s = \sum_{x, e \neq 0} \Gamma(x, e) p(x, e) \quad (5.14)$$

donde los coeficientes $\Gamma(x, e)$ tienen valores que dependen de las funciones que implementa el módulo de extracción; concretamente

$$\Gamma(x, e) = \begin{cases} 1, & \text{si las salidas del extractor para } x \text{ y para } xoe \text{ difieren.} \\ 0, & \text{en caso contrario.} \end{cases}$$

y $p(x, e)$ es la probabilidad de que al aplicar una combinación de entradas al circuito a testear se obtenga la salida xoe en lugar de la salida x como en el circuito sin defectos. Es decir, es la probabilidad de que la salida x se vea afectada por el patrón de error e .

Antes de evaluar p_s para un circuito de extracción dado hay que obtener los valores de los $p(x, e)$. El conocimiento de esas probabilidades implica determinar qué salidas se obtienen para cada uno de los patrones de test y qué patrones de error se producen cuando el circuito se ve afectado por cualquiera de los defectos posibles cuando se aplica un patrón de test.

Aunque, por su complejidad, no se disponga de una expresión algebraica de la función que implemente el circuito, mediante simulación lógica se puede conocer qué salidas se obtienen para cada patrón de test en el circuito correcto. Esto no supone un problema grave si el número, L , de patrones de test no es muy grande. Sin embargo, la determinación de la distribución de patrones de error para todos y cada uno de los posibles defectos supone un

serio obstáculo. Ello es debido, por una parte, a que se necesita disponer de modelos de defectos (faltas) adecuados y fáciles de programar de cara a poder realizar una simulación rápida y, por otra, a que el número de posibles defectos a considerar puede ser excesivamente elevado. Baste considerar que, en un circuito que se describa en el nivel de puertas lógicas como un circuito de N nudos y en el que, por consiguiente, se utilice el modelo de faltas de anclaje de líneas para modelar los defectos, habría que simular $3^N - 1$ faltas.

Se han planteado diversas posibilidades para eludir estas dificultades, vistas con detalle en la sección 1.2. Entre estas posibilidades, destacan:

- (a) Simular todas las faltas de anclaje simples.
- (b) Simular una muestra significativa de faltas de anclaje simples y múltiples.
- (c) Realizar ciertas hipótesis sobre el tipo de distribuciones de patrones de error que se tendrán, a partir de los defectos que pueden producirse y de la estructura del circuito.

En algunos trabajos recientes [KAR90], se presenta una forma de calcular pd bastante inmediata. Para que se pueda utilizar es preciso que se verifiquen las siguientes hipótesis:

- (i) Las probabilidades de los errores y las de las salidas son independientes.
- (ii) Todos los patrones de error son equiprobables.
- (iii) Todas las salidas pueden producirse con igual probabilidad.

Una vez conocidas o supuestas la distribución de errores posibles en las salidas del circuito, aún queda la determinación de la función $\Gamma(x,e)$ para la que se alcanza un valor de ps suficientemente alto. Precisamente, cuanto mayor sea el valor de ps para un módulo extractor, mayor es su efectividad. La obtención de ps se estudia con detalle en el capítulo 6, donde se considera el diseño del módulo extractor de los circuitos testeables concurrentemente.

5.4.El procedimiento de test concurrente propuesto.

En la figura 5.7 se muestra el esquema de bloques de los circuitos testeables concurrentemente según el diseño que proponemos. En esa figura se pueden distinguir los tres módulos a añadir al circuito a testear:

- (a) Módulo extractor.
- (b) Módulo predictor y monitor de patrones.
- (c) Módulo comparador.

Comparando con el circuito de la figura 5.5, se tiene que los módulos extractor y comparador corresponden al módulo de comprobación de código, CC, y el módulo predictor y monitor de patrones corresponde a la circuitería que genera los bits de código (δX) para las salidas normales del circuito.

El módulo extractor genera la función de extracción, $G(X)$, que actúa sobre las salidas del circuito, X , de manera que se obtenga una cierta propiedad que posean en el caso de que sean correctas. El módulo predictor realiza la función de predicción, $H(Y)$, que a partir de cada entrada al circuito que sea considerada patrón de test, genera la propiedad que ha de mostrar la salida correspondiente. Por último, el circuito comparador determina si la propiedad extraída coincide con la predicha y si no es así indicará que ha habido error. Es decir, si $F_1=(F,H)$ es la función que sintetiza el circuito modificado y F la del circuito original, cuando $G(F(Y)) \neq H(Y)$ se ha producido error. Este sistema será autocomprobante para un conjunto de faltas FL, si:

$$\forall f \in FL \text{ existe } Y / H(Y) \neq G(F_1(Y)) \quad (5.15)$$

y será seguro frente a las faltas de FL si

$$\forall f \in FL \text{ cuando } F(Y) \neq F^*(Y) \text{ entonces } G(F^*(Y)) \neq H(Y) \quad (5.16)$$

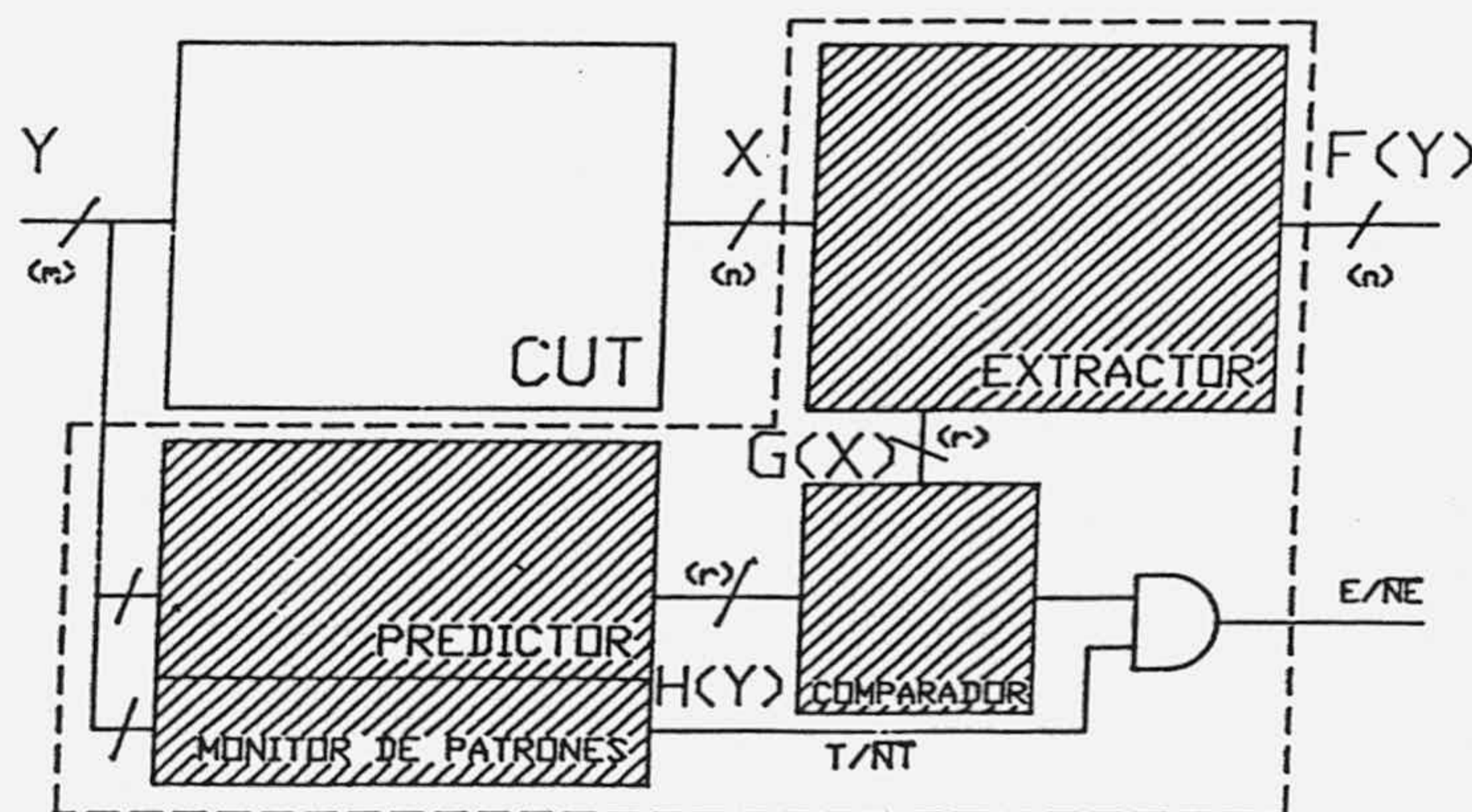


Figura 5.7. Módulos añadidos a un circuito para el test propuesto.

Para poder asegurar (5.15) es preciso que exista algún patrón de test para cada falta de F , y para asegurar (5.16) se necesita que la probabilidad de enmascaramiento de la función de compresión asociada al circuito extractor sea cero. En lo que sigue, evaluaremos la capacidad del método de detección propuesto en términos de la probabilidad de que se produzca un error en la

salida del CUT debido a una falta, p_e , y de la probabilidad de enmascaramiento, p_A , según la expresión 5.9.

En el esquema que se ha propuesto se añaden al circuito a testear, CUT, los módulos que constituyen el circuito comprobador, CC, cuya función global es detectar defectos en el CUT. Sin embargo surge la cuestión de la detección de errores en el CC. Para responder a este problema hay que partir de la hipótesis de que en el circuito testeable concurrentemente existe muy poca probabilidad de que se produzcan defectos múltiples. De esta forma se va a presuponer que sólo se puede tener un defecto en el CUT o un defecto en el CC.

La situación que se ha tenido en cuenta hasta ahora corresponde a un defecto en el CUT. Si el defecto está en el CC, lo más que puede suceder es que se dé por errónea una salida del CUT que no lo es. Entonces, la señal de error no indica error en el CUT sino en el CC. En ambos casos, si se indica error es porque existe un defecto, ya sea en el CUT o en el CC.

En otros trabajos [BOS84d, HUG84, JHA89, KHA84, PIE87, TAM84] sí que se considera un CC autocomprobante, concretamente, los estudios de autocomprobabilidad se refieren principalmente al módulo comparador. Considerar esta posibilidad es imprescindible sobre todo si el circuito comprobador no se encuentra en el mismo circuito integrado que el CUT, en cuyo caso es discutible que los defectos sólo afectan al CUT o al CC.

Así pues, el procedimiento que se presenta aquí permite codificar las salidas utilizando el circuito predictor para algunas de las entradas y considerando la posibilidad de detección que ofrece el módulo extractor y el módulo comparador diseñados previamente teniendo en cuenta los patrones de error posibles y la máxima complejidad. En ese sentido se puede considerar un procedimiento Tipo I. Sin embargo, sólo para ciertas entradas (los patrones de test) se habilita el circuito comprobador para que analice si la salida es correcta o no. Con ello se pretende disminuir la complejidad de los circuitos testables concurrentemente. En comparación con los procedimientos de Tipo II, las entradas para las que se activa la circuitería de comprobación no han de seguir una secuencia preestablecida y, por consiguiente, se podrán conseguir tiempos de latencia inferiores a los de los procedimientos Tipo II. Además, para la selección de los patrones de test se pueden utilizar las mismas herramientas disponibles para el test off-line, en particular se puede usar el procedimiento presentado en el capítulo 3 de la presente memoria. Consiguientemente, este procedimiento puede ser ventajoso en cuanto a la simplicidad de la circuitería a añadir, consiguiendo tiempos de latencia aceptables, aunque quizá mayores que en el caso de los procedimientos Tipo I.

5.4.1. Análisis y síntesis de circuitos testeables concurrentemente.

En la figura 5.8 se esquematizan el proceso de análisis y el proceso de síntesis para circuitos testeables concurrentemente según el esquema propuesto en la presente memoria. Las elipses de la figura representan los distintos módulos del circuito, es decir, el circuito de partida que hay que testear (notado CUT), el módulo extractor (extractor), el módulo de predicción (predictor) y el módulo monitor de patrones (monitor). Los recuadros indican los parámetros que se utilizan para evaluar la efectividad del método de test concurrente; así, por una parte están la probabilidad de detección (p_s) y la probabilidad de aplicar un patrón de test que origine una salida errónea (p_e), utilizadas para establecer el tiempo de latencia medio (TL), y por otra el incremento de complejidad que supone la circuitería añadida (δC).

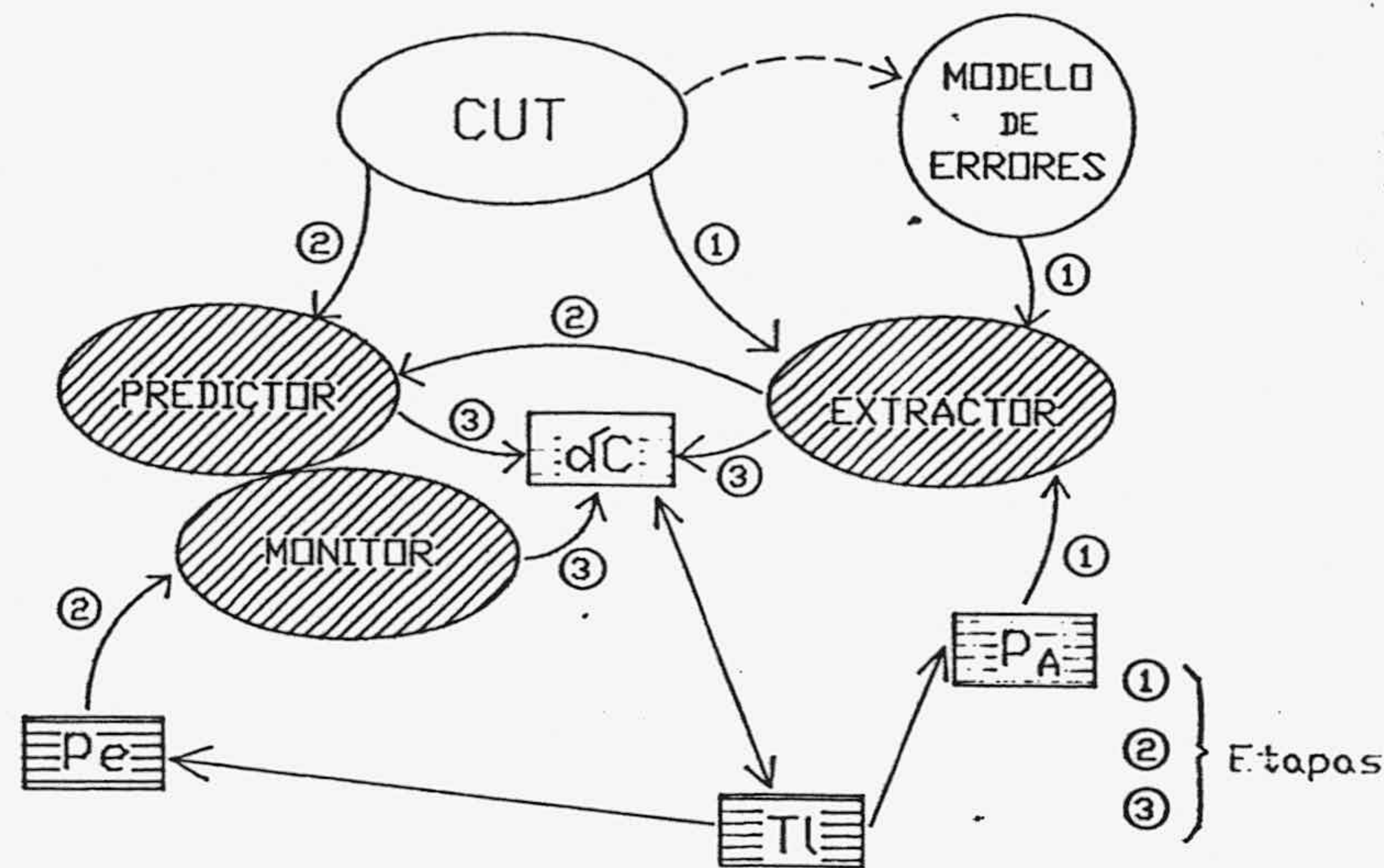


Figura 5.8. Esquema para la síntesis y el análisis de un circuito testeable según el procedimiento propuesto.

Considérese en primer lugar el análisis de la efectividad del circuito testeable concurrentemente. A partir de la distribución de probabilidades de error en las salidas del circuito y teniendo en cuenta las funciones que sintetizan el circuito, el módulo predictor y el módulo extractor, se puede determinar la probabilidad de detección, p_s . Utilizando las entradas que el módulo monitor de patrones selecciona como patrones de test y la probabilidad de que se produzca un error para estos patrones, se obtiene la probabilidad de error en las salidas del CUT, p_e . A partir de la probabilidad de detección y de la probabilidad de error se calcula el tiempo de latencia medio y, por último, conociendo la realización de los módulos predictor, extractor y monitor de patrones se evalúa la complejidad de la circuitería añadida.

En el caso de la síntesis de los módulos que hacen al circuito testeable concurrentemente, el objetivo es determinar las funciones que han de implementar los módulos extractor, predictor y monitor de patrones de forma que con la circuitería adicional más simple posible se minimice el tiempo de latencia. Hay que tener en cuenta que el tiempo de latencia y la complejidad del hardware añadido son magnitudes ligadas de tal forma que la optimización de una se hace a costa de la otra, siendo necesario llegar a un compromiso en la optimización de ambas: si sólo interesara optimizar el tiempo de latencia, la solución se obtiene de forma inmediata, ya que bastaría con duplicar o triplicar el circuito a testear y añadir un módulo que comprobara la coincidencia de las salidas.

El procedimiento seguido para realizar la síntesis se estudia en los dos capítulos siguientes y comprende los siguientes pasos:

- (a) **Especificación del tiempo de latencia máximo admisible.**
- (b) **Especificación de la complejidad máxima.** Esta no debería ser superior a la que implicaría duplicar el CUT.
- (c) **Síntesis del módulo extractor** que optimiza la probabilidad de enmascaramiento y que tiene una complejidad menor que un valor dado. Para llevar a cabo la síntesis se tiene en cuenta la distribución de las probabilidades de error del circuito a testear.
- (d) **Obtención de la probabilidad de detección mínima** a partir de la probabilidad de enmascaramiento del módulo extractor obtenido en (c).
- (e) **Selección del conjunto de patrones de test** para los que se consigue la probabilidad de detección requerida con la complejidad mínima para el circuito monitor de patrones.
- (f) **Síntesis mínima del módulo predictor y del módulo monitor de patrones.** Para ello se tienen en cuenta las funciones que implementa el circuito y el módulo extractor para las entradas que son patrones de test y se consideran indiferencias en las entradas que no lo son.
- (g) **Evaluación de la complejidad de la circuitería añadida.** Esta debe ser menor que la que implicaría duplicar el circuito a testear.

De los pasos anteriores, el (a), el (b) y el (d) no necesitan comentario alguno. En cuanto a los restantes, el (c) se estudia en el capítulo 6, y el capítulo 7 se dedica a las cuestiones que plantean los pasos (e) y (f). El paso (g) se considera en el capítulo 8 a partir de los ejemplos de circuitos a los que se les ha aplicado el procedimiento propuesto para diseñar circuitos testables concurrentemente.

Hasta ahora no se ha dicho nada acerca de la síntesis del módulo

comparador. Esta debe realizarse de manera que este módulo sea seguro frente a fallos y, a tal fin, puede consultarse una gran cantidad de trabajos publicados al respecto. Entre estos se pueden destacar [BOS84d, HUG84, JHA89, PIE87, TAM84].

5.5. Conclusión.

En este capítulo se han presentado los tópicos relativos al test concurrente de circuitos integrados. Se ha introducido la terminología utilizada en el contexto de los sistemas tolerantes a fallos y se ha elaborado un modelo para analizar la efectividad de tales sistemas.

También se ha mostrado la relación entre el estudio de los sistemas tolerantes a fallos y la necesidad del test concurrente. Así se ha relacionado la capacidad de detección de fallos y, por consiguiente, la fiabilidad en los sistemas tolerantes a fallos, con el tiempo medio de latencia en un sistema testeable concurrentemente.

Se han descrito los módulos que se añaden a un circuito para hacerlo testeable concurrentemente y se han establecido las condiciones que deben cumplir. Además se han presentado parámetros que permiten cuantificar la efectividad del test concurrente, como el tiempo de latencia, TL , la probabilidad de error en las salidas, p_e , y la probabilidad de detección, p_s . Estos parámetros permiten dirigir el diseño de los módulos para el test concurrente.

El procedimiento de test concurrente propuesto permite una complejidad menor para la circuitería añadida que en los procedimientos basados en la codificación de las salidas, y un tiempo de latencia menor que los procedimientos que aprovechan la circuitería del test BIST off-line. Supone, por consiguiente, un punto intermedio, aparte de permitir una cierta aproximación gradual a un menor tiempo de latencia a costa de aumentar el hardware o a una menor complejidad del hardware mediante un aumento del tiempo de latencia.

El diseño de los módulos a añadir se considera en los capítulos 6 y 7, dándose en el capítulo 8 resultados de la aplicación a algunos circuitos del proceso descrito.

CAPITULO 6. DISEÑO DEL MODULO EXTRACTOR.

En el capítulo 5 se describen los módulos de un circuito testeable concurrentemente y se presentan las expresiones que permiten determinar la efectividad del diseño de test concurrente que se trata de realizar.

En el presente capítulo se desarrolla un procedimiento de diseño del módulo extractor para optimizar (maximizar) el valor de la probabilidad de detección de un error a la salida, ps . Puesto que cualquier función se puede expresar de forma unívoca mediante el espectro de Tamari, el problema se traduce en la búsqueda del conjunto óptimo de coeficientes espectrales que definan las funciones sintetizadas por el módulo extractor. La expresión (5.14) da ps en función de $p(x,e)$ y de $\Gamma(x,e)$. Por consiguiente, se trata de determinar la función que debe sintetizar el módulo extractor para que $\Gamma(x,e)$ maximice ps dada una distribución de patrones de error.

En la sección 6.1 se muestra cómo el módulo extractor de un circuito testeable concurrentemente no es más que un módulo compresor de salidas y, por consiguiente, se puede analizar desde el mismo punto de vista, es decir, utilizando la probabilidad de enmascaramiento, P_A . La sección 6.2 se dedica a la obtención de la probabilidad de enmascaramiento asociada a un módulo extractor en términos de los coeficientes espectrales de Tamari. En la sección 6.3 se estudia la obtención de los modelos de error necesarios para evaluar el enmascaramiento. La sección 6.4 describe un procedimiento para la determinación de los coeficientes espectrales que definen el módulo extractor.

6.1. El módulo extractor como compresor de salidas.

El módulo extractor realiza una modificación de las salidas del circuito cuando se aplica alguna de las entradas de test. En este sentido, su función es la misma que la de un circuito de compresión de las salidas en un procedimiento BIST para una secuencia de entradas al circuito de compresión formada por una única entrada ($L=1$). Abundando en esta identificación, los módulos de predicción y de monitorización de patrones se encargarían de generar la signatura para cada patrón de test que se compare con la salida del módulo extractor. Según esto, el test concurrente se puede considerar un caso particular del test BIST en el que la longitud de la secuencia de test es 1 y se incorpora además la circuitería que proporciona la signatura del circuito correcto. Por consiguiente, utilizando la expresión de la probabilidad de enmascaramiento (1.21), para $L=1$ se tendrá que:

$$P_A = \sum_{x, e \neq 0} A(X^T, E^T) P(X^T, E^T) = \sum_{x, e \neq 0} A(x, e) p(x, e) \quad (6.1)$$

en donde se han sustituido las secuencias X^T y E^T de un sólo elemento por x y e respectivamente, siendo $x=(f_1(Y), \dots, f_N(Y))$ y $e=(e_1, \dots, e_N)$.

Teniendo en cuenta la definición (5.14) de la función Γ , se puede definir una función de enmascaramiento, A , como sigue:

$$A(x, e) = 1 - \Gamma(x, e) \quad (6.2)$$

y sustituyendo en (6.1) se tiene

$$P_A = \sum_{x, e \neq 0} A(x, e) p(x, e) = \sum_{x, e \neq 0} (p(x, e) - \Gamma(x, e) p(x, e)) = 1 - p_s \quad (6.3)$$

Así, la expresión (6.3) establece la relación entre la probabilidad de enmascaramiento debida al módulo compresor y la probabilidad de detección que proporciona. Maximizar p_s significa, por consiguiente, minimizar P_A .

El resto del capítulo está dedicado a la búsqueda de las funciones que ha de sintetizar el módulo extractor para que el valor de P_A sea el menor posible. El diseño de compresores óptimos que proporcionen un cubrimiento similar para todos los patrones de error es un problema abierto [KAR90].

6.2. Efectividad del extractor en términos de los coeficientes de Tamari.

En este apartado, se expresará la probabilidad de detección del módulo

extractor, ps , de forma que aparezcan explícitamente los coeficientes del espectro de Tamari. Estos definen la función que ha de realizar dicho módulo (función de extracción) y permiten su síntesis. Partiendo de

$$ps = \sum_{x, e \neq 0} \Gamma(x, e) p(x, e) \quad (6.4)$$

se pretende determinar la función de extracción que maximice ps para una distribución de errores dada. La función buscada se notará como $G(x) = (g_1(x), \dots, g_r(x))$, donde $g_i \in F_2^n$ ($i=1, 2, \dots, r$) y $x = (x_1, \dots, x_n)$ con $x_j \in C_2$.

En la figura 5.7 se muestran los módulos que se añaden a un circuito (CUT) para hacerlo testeable concurrentemente. Entre esos módulos se encuentra el módulo extractor. Sus entradas, $x \in C_2^n$, corresponden a las salidas del CUT y el número de funciones que sintetiza es r . Interesa que r sea lo más bajo posible a fin de reducir la complejidad del módulo de extracción.

El módulo extractor no podrá establecer la presencia de un error, e , para una salida prevista del circuito correcto, x , cuando se verifique que

$$g_i(x) \oplus g_i(x \oplus e) = 0 \quad \forall i=1, \dots, r \quad (6.5)$$

Utilizando (6.5) se puede definir una función de enmascaramiento, A_i , y la correspondiente probabilidad de enmascaramiento por salida del módulo extractor, p_{A_i} , que tendrá la forma:

$$p_{A_i} = \sum_{x, e \neq 0} p(x, e) A_i(x, e) \quad (6.6)$$

con $A_i(x, e) = g_i(x) \oplus g_i(x \oplus e)$

Además, a partir de $A_i(x, e)$ se puede definir la correspondiente función de detección Γ_i , verificando que

$$\Gamma_i(x, e) = 1 - A_i(x, e) = 1 - (g_i(x) \oplus g_i(x \oplus e)) \quad (6.7)$$

y, sustituyéndola en (6.6)

$$p_{A_i} = 1 - \sum_{x, e \neq 0} p(x, e) (g_i(x) \oplus g_i(x \oplus e)) \quad (6.8)$$

Como las funciones g_i que constituyen el módulo extractor son independientes, la probabilidad de enmascaramiento p_A se obtiene de los valores de las p_{A_i} a partir de

$$p_A = \prod_i p_{A_i} \quad (i=1, 2, \dots, r). \quad (6.9)$$

La probabilidad de detección se obtiene directamente de p_A con (6.3). Así pues, el objetivo que se persigue se concreta es minimizar (6.9),

intentando además minimizar la complejidad del circuito que implementa el módulo extractor: el número de funciones, g_i , distintas debe ser el menor posible (es decir hay que minimizar P_A y r). Teniendo esto en cuenta, el proceso a seguir se concreta en el algoritmo siguiente:

Algoritmo 6.1:

1. Hacer $r=1$, $i=1$.
2. Buscar la función g_i que minimiza (6.8).
3. Obtener el valor de p_{Ai} correspondiente a la g_i solución.
4. Evaluar (6.9) utilizando p_{A1}, \dots, p_{Ar} .
5. Si p_A es suficientemente baja finalizar, si no ir a 6.
6. Incrementar r e i . Ir a 2.

El núcleo de este algoritmo es el paso 2. Se trata de encontrar la función g_i que minimiza (6.8). Para concretar el procedimiento de búsqueda es preciso expresar (6.8) de forma operativa: aquí se han utilizado los coeficientes de Tamari. De esta forma en el paso 2 se busca el espectro de la función g_i que minimiza (6.8), que a su vez define el circuito de extracción. La expresión de la función g_i en términos de los coeficientes de Tamari es:

$$g_i(x) = \sum_j \alpha_j T_j(x) \quad (6.10)$$

de manera que si se produce un error, $e=(e_1, \dots, e_n)$, se tendrá

$$g_i(x \oplus e) = \sum_{j=0}^N \alpha_j T_j(x \oplus e) \quad (6.11)$$

siendo

$$T_j(x) = x_1^{j_1} x_2^{j_2} \dots x_n^{j_n} \quad (6.12)$$

donde $x=(x_1, \dots, x_n)$ y $j=(j_1, \dots, j_n)$

α_j el coeficiente espectral j -ésimo, y $N=2^n-1$.

Resultará:

$$T_j(x \oplus e) = (x_1 \oplus e_1)^{j_1} \dots (x_n \oplus e_n)^{j_n} \quad (6.13)$$

Además, utilizando las propiedades de las funciones de Tamari en F_2^n :

$$\begin{aligned} (1) \quad T_1(x) &= 1 \text{ si } l_j \leq x_j \quad \forall j=1, \dots, n \\ &= 0 \text{ en otro caso} \\ (2) \quad T_1(x \oplus e) &= 1 \text{ si } l_j \leq x_j \oplus e_j \quad \forall j=1, \dots, n \\ &= 0 \text{ en otro caso} \end{aligned} \quad (6.14)$$

se verifica que

$$g_i(x) \oplus g_i(x \oplus e) = \sum_{k=0}^N \alpha_k (T_k(x) \oplus T_k(x \oplus e)) = \sum_{k \in S(x,e)} \alpha_k \quad (6.15)$$

donde el conjunto $S(x,e)$ contiene todos los índices que verifiquen:

$$\begin{aligned} (a) \quad & k_j \leq x_j \quad \forall j=1, \dots, n \\ & \text{ó} \\ (b) \quad & k_j \leq x_j \oplus e_j \quad \forall j=1, \dots, n, \quad \text{pero no (a) y (b) a la vez.} \end{aligned} \quad (6.16)$$

Sustituyendo (6.15) en (6.8) se obtiene la expresión de la probabilidad de enmascaramiento para la salida i -ésima del módulo extractor en términos de sus coeficientes de Tamari.

$$p_{Ai} = 1 - \sum_{x,e \neq 0} p(x,e) [\sum_{j \in S(x,e)} \alpha_j] \quad (6.17)$$

Como ejemplo, sean $x=(0,1,1)$, $e=(1,0,1)$. Los índices j que verifican la condición (a) de (6.16) tienen la forma $(0,d,d)$ donde d es una indiferencia. Los índices j que verifican la condición (b) de (6.16) tendrán la forma $(d,d,0)$ ya que $x \oplus e = (1,1,0)$. Así, los que sólo verifican una de las dos condiciones son $(0,d,1)$ y $(1,d,0)$; es decir

$$S((0,1,1), (1,0,1)) = \{(0,0,1), (0,1,1), (1,0,0), (1,1,0)\}$$

A partir de este ejemplo, se puede entender el siguiente algoritmo de obtención de los coeficientes de Tamari que intervienen en la suma EXOR para una pareja, x y e , dada:

Algoritmo 6.2:

1. Dado el valor de x construir el vector $v1$ que tenga ceros en aquellas posiciones donde x tenga ceros y d en las otras posiciones.
2. Obtener $x \oplus e$ y construir el vector $v2$ que tenga ceros en aquellas posiciones donde $x \oplus e$ tenga ceros y d en las otras posiciones.
3. Determinar los índices contenidos en $v1$ y en $v2$ al mismo tiempo. Con ellos se construye el conjunto $v1 \cap v2$.
4. El conjunto $S(x,e)$ está definido por las índices contenidos en $v1$ y en $v2$ pero que no estén en $v1 \cap v2$.

■

Falta por determinar aquellas parejas, (x,e) , para las que cada coeficiente de Tamari, α_j , interviene en la suma EXOR que multiplica al correspondiente $p(x,e)$. Esto se puede conseguir aprovechando la condición (6.18) para establecer el siguiente algoritmo:

Algoritmo 6.3:

1. Construir los conjuntos $C(j)$ y $C'(j)$ definidos como $C(j) = \{x \in C_2^n / j \subseteq x\}$,

$C'(j) = \{y \in C_2^n / j \notin y\}$ (se puede ver que $C(j) \cup C'(j) = C_2^n$).

2. Obtener los productos cartesianos $C(j) \times C'(j)$ y $C'(j) \times C(j)$.

3. Por cada elemento $(x, y) \in C(j) \times C'(j)$ se obtiene una pareja (x, e) con $e = x \oplus y$ para la que α_j multiplica a $p(x, e)$. (Obsérvese que, puesto que $x \neq y$, siempre $e \neq 0$).

4. Por cada elemento $(y, x) \in C'(j) \times C(j)$ se obtiene una pareja (y, e) con $e = x \oplus y$ para la que α_j multiplica a $p(y, e)$.

■

Si se nota con $N(j)$ el número de unos que contiene j , el número de elementos de $C(j)$ es $2^{N(j)}$ y el número de elementos de $C'(j)$ es, por consiguiente, $2^N - 2^{N(j)}$. Así, el número de parejas distintas de patrones de entrada al módulo extractor y patrones de error es $2(2^N - 2^{N(j)})(2^{N(j)})$.

Con esto se completa la caracterización de la ecuación (6.8) en términos de los coeficientes espectrales de Tamari. Ahora, el problema es encontrarlos.

Como se demuestra a continuación, la búsqueda de los coeficientes espectrales que optimizan (6.17) para una determinada distribución de probabilidades de error constituye un problema NP-Completo.

Teorema 6.1: Encontrar el conjunto de coeficientes de Tamari que minimizan (6.17) es un problema NP-completo.

Demostración: Saber que el valor p_A' es el valor mínimo para (6.17), una vez dada la distribución de probabilidades $p(x, e)$, implica conocer un conjunto de coeficientes, $a(x, e)$, iguales a cero o a uno para los cuales se verifican las igualdades $\sum'_{j \in S(x, e)} \alpha_j = a(x, e)$, que hacen $p_A' = 1 - \sum_{x, e \neq 0} p(x, e) a(x, e)$. Es decir, se ha de resolver la ecuación booleana

$$\bigcup_{x, e \neq 0} (a(x, e) \oplus \sum'_{j \in S(x, e)} \alpha_j) = 1 \quad (6.18)$$

lo cual es un problema NP-Completo [IBA75, FUJ82, FUJ90].

■

En el caso concreto que se está considerando, la complejidad sería del orden de 2^{2^n} , ya que ese es el número de posibles formas de seleccionar los 2^n coeficientes espectrales que definirían la función de extracción.

En la sección 6.4 se considera la determinación de los coeficientes de Tamari para optimizar (6.17) mediante una aplicación del algoritmo de **Recocido Simulado** [CHA88, JAY88, KIR83, OTT88]. Previamente, en la sección 6.3 se estudia la obtención del modelo de errores para el circuito a testear. En el apéndice V se estudia la posibilidad de resolver el problema de optimización de (6.17) utilizando una **red neuronal de Hopfield Generalizada (GHN)** [HOP82, 84, TSI89]

6.3. Modelos de error.

Como se ha visto, para evaluar la probabilidad de enmascaramiento se necesita conocer la forma en la que se manifiestan los defectos en un circuito dado: se ha de disponer de un modelo de error para dicho circuito.

En el capítulo 1 se muestra la necesidad de **modelar los defectos de un circuito mediante faltas** para así determinar qué combinaciones de entradas los ponen de manifiesto, sin prestar atención a cual sea el patrón de error que se obtenga, únicamente interesa que sea distinto de cero. Sin embargo, de cara al análisis de la probabilidad de enmascaramiento de un método de compresión, resulta imprescindible conocer algo respecto a los patrones de error que se puedan observar. Esa información la dan los **modelos de error**.

Supóngase el circuito R_F que sintetiza las n funciones $F=(f_1, \dots, f_n)$, $f_i \in F_2^m$, con entradas $y=(y_1, \dots, y_m)$, $y_i \in C_2^m$. Dado el conjunto de todos los posibles defectos que puedan presentar, $D=\{d_1, d_2, \dots, d_M\}$, para cada uno de esos defectos, d_i , se define una función característica σ_{d_i} como sigue:

$$\sigma_{d_i}(y, e) = \begin{cases} 1 & \text{si } F_{d_i}(y) = e \oplus F(y) \\ 0 & \text{en caso contrario} \end{cases} \quad \forall y \in C_2^m, e \in C_2^n \quad (6.19)$$

donde F_{d_i} son las funciones que sintetiza el circuito cuando se ve afectado por el defecto d_i . A partir de ellas se puede evaluar la probabilidad $p(e/y)$ de tener un patrón de error e dada una combinación de entradas y , como sigue:

$$p(e/y) = \sum_{d_i \in D} \sigma_{d_i}(y, e) p(d_i) \quad (6.20)$$

donde con $p(d_i)$ se denota la probabilidad de que se produzca el defecto d_i .

Así pues, para determinar la distribución de probabilidades de errores que aparece en (6.17), se pueden utilizar (6.19) y (6.20) ya que la probabilidad de que se produzca una salida x en lugar de una salida x viene dada por:

$$p(e/x) = \sum_y p(e/y) p(y/x) \quad (6.21)$$

donde la sumatoria se extiende a todas las combinaciones de entradas, y , para las que la salida correcta del circuito es x , es decir $F(y)=x$. Para evaluar (6.21) se precisaría conocer las entradas del circuito que dan lugar a una salida determinada. No obstante, en el caso de que se haya fijado una secuencia de patrones, sólo habrá que tener en cuenta los patrones del test

que producen cada salida.

Una vez obtenidos los valores de las probabilidades condicionadas, $p(e/x)$, la probabilidad de obtener una salida x errónea, siendo e el patrón de error es:

$$p(x,e) = \sum_x p(x) p(e/x) \quad (6.22)$$

Ahora bien, la evaluación de (6.22) plantea problemas que resultan más insalvables cuanto mayor es la complejidad del circuito:

(a) El número de posibles defectos a considerar es muy elevado. Normalmente ni siquiera se puede elaborar una lista exhaustiva de ellos o no se dispone de información de cuales pueden ser. Aún si se hace la suposición de que los defectos más significativos son aquellos que se modelan mediante faltas de anclaje o de stuck-open simples, el número de faltas a considerar puede ser muy elevado, como se ha visto en la sección 1.2.3.

(b) Hay que determinar qué patrón de error se observa para cada uno de los defectos y para cada entrada. Si el número de entradas es elevado, puede ser imposible considerar todas y cada una de ellas.

Para evitar estas dificultades, existen dos caminos posibles:

(i) Se simula el comportamiento del circuito para una muestra de defectos utilizando el conjunto de patrones de test seleccionados. Los resultados se pueden justificar a partir del teorema central del límite [PHE84], que permite establecer el tamaño de la muestra de entradas y defectos.

(ii) Se presupone un modelo de errores para el circuito, es decir, se establece una serie de hipótesis acerca de las características de las probabilidades $p(x,e)$, $\forall x=F(y) \in C_2^n$ y $e \in C_2^n$. Según los modelos de error que se han establecido en la literatura [ABR86b], se puede hablar de errores **simétricos** y de errores **unidireccionales**. Los errores simétricos son aquellos en los que, dada una salida $x=(x_1, \dots, x_n)$, se puede producir un error en cualquier bit, x_i , independientemente de que $x_i=0$ ó $x_i=1$. En cambio, en los errores unidireccionales, dada la salida x , sólo se pueden producir errores en aquellos $x_i=0$ o sólo en los $x_i=1$, con lo que el número de unos en x es siempre distinto del número de unos en $x \oplus e$. Así por ejemplo, en las PLAs, los defectos de contacto (crosspoint) simples se traducen en errores unidireccionales. Según el número de bits erróneos que se puedan tener se habla de errores simples o múltiples.

A continuación, en la sección 6.3.1 se describe el método que se ha seguido para obtener las probabilidades de los patrones de error mediante el muestreo de defectos.

6.3.1. Obtención experimental de un modelo de error.

En esta sección se describe un procedimiento de construcción de las distribuciones de probabilidad de errores para las distintas combinaciones de salidas del circuito. Esas distribuciones son esenciales para obtener una medida realista de la probabilidad de enmascaramiento del procedimiento de compresión elegido. No obstante, si no se dispusiera de ellas, se podría hacer alguna hipótesis acerca del tipo de patrones que se pueden encontrar, es decir, presuponer un modelo de errores. Esa hipótesis se contrastaría después con los resultados experimentales obtenidos.

Para determinar las probabilidades $p(x,e)$ se supone que: (a) el circuito de n salidas se comporta de manera similar a n fuentes de error binarias independientes [WIL88] (figura 6.1); o bien (b) el circuito de n salidas se comporta como una fuente de error q -aria, con $q=2^n$ [PRA90], como la de la figura 6.2. Tanto en un caso como en otro, las probabilidades $p(x,e)$ se obtendrán a partir de las frecuencias de cada patrón de error y cada salida. Para eso:

- (i) Se modelan los defectos mediante faltas,
- (ii) Se simula el comportamiento del circuito afectado por ellas, y
- (iii) Se contabilizan los patrones de error que se presentan en las salidas y se calculan las correspondientes frecuencias.

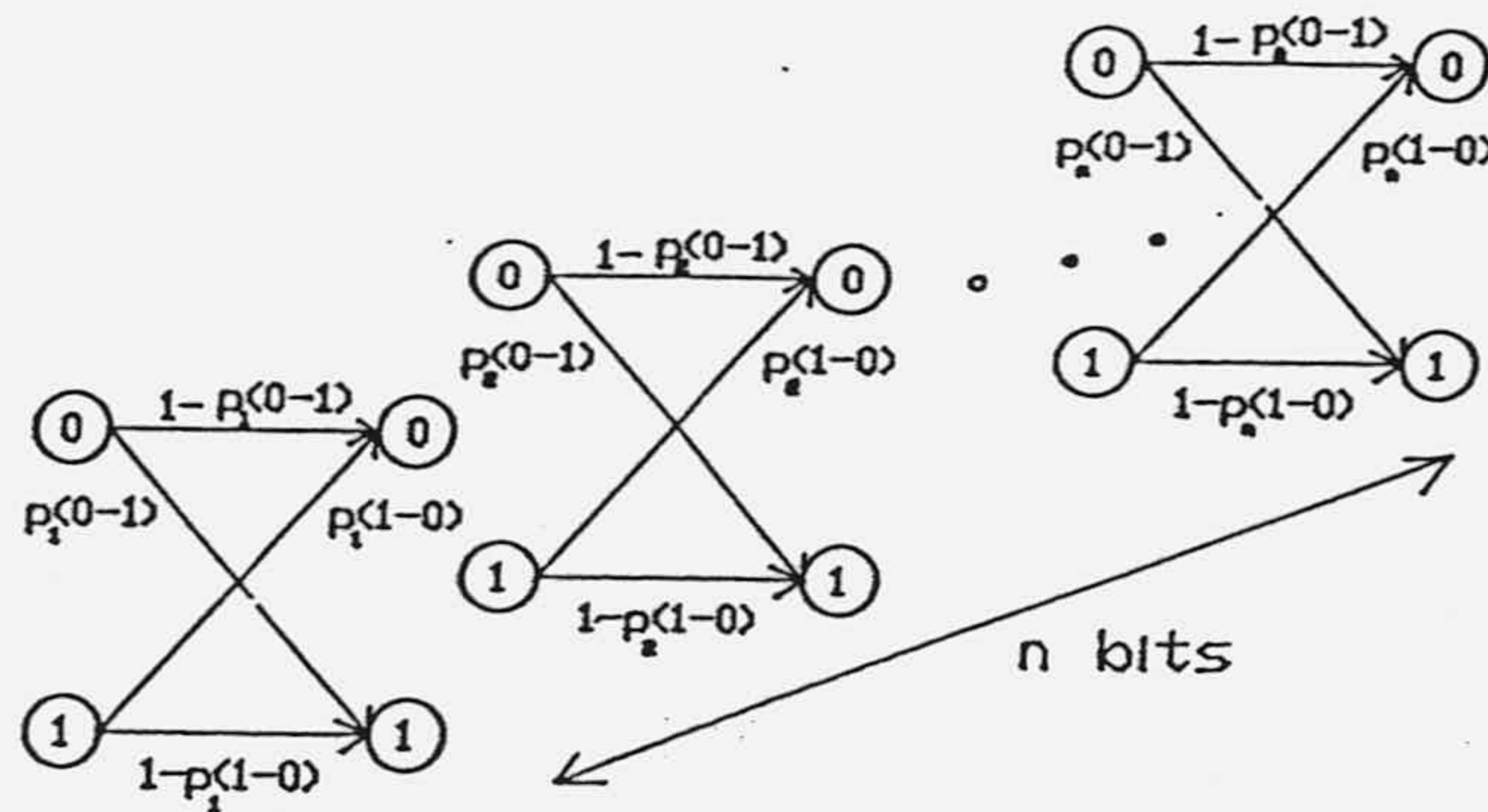


Figura 6.1. Modelo de error independiente para cada salida mediante un canal binario (simétrico o asimétrico).

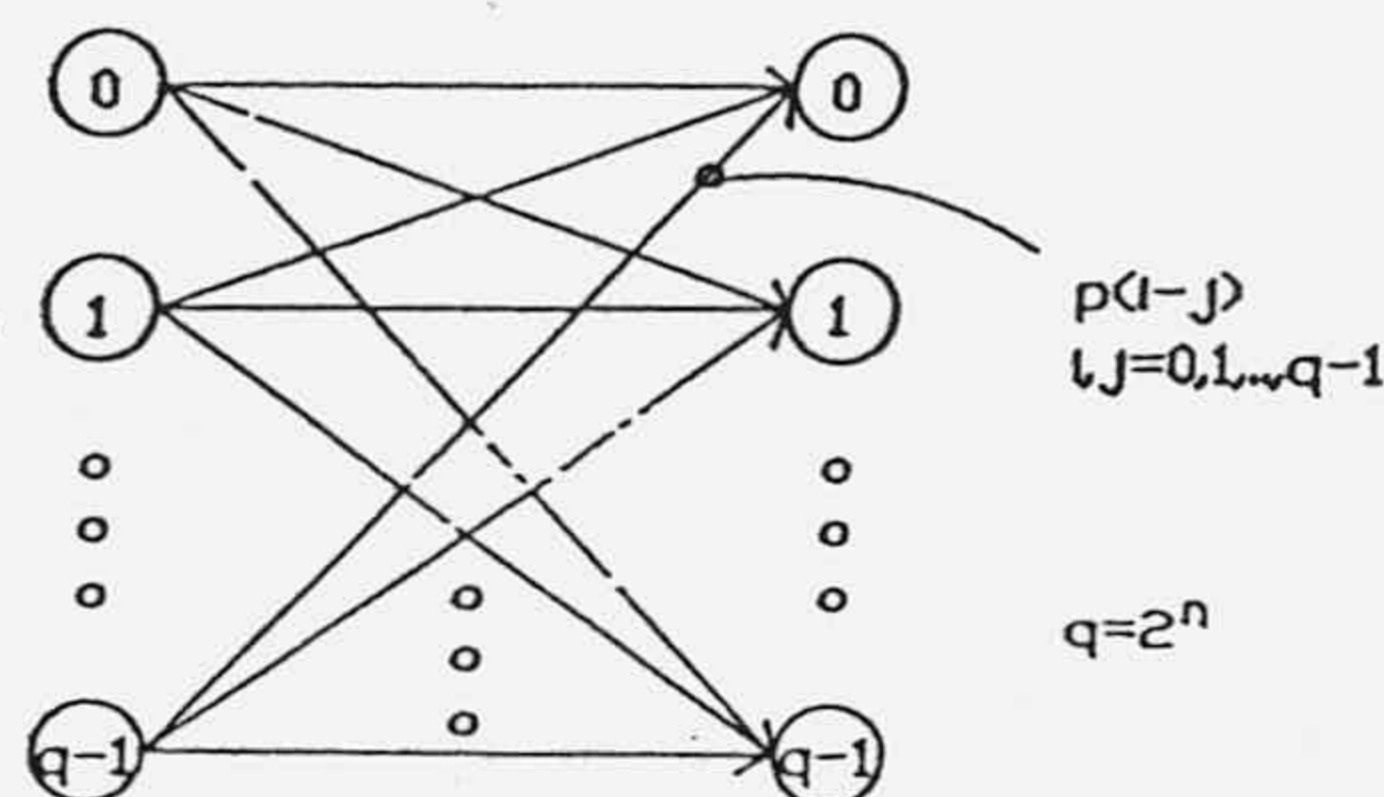


Figura 6.2. Modelo de error para cada combinación de salida con canal q -ario.

Sin embargo, este planteamiento es poco menos que imposible en el caso de circuitos complejos, como los circuitos VLSI, ya que precisa de una simulación exhaustiva de todas las faltas posibles y asignar a dichas faltas una cierta probabilidad de que se produzcan. Así, aunque sólo se utilizasen faltas de anclaje simples para modelar los defectos de un circuito con G puertas lógicas, el tiempo de simulación (en el caso de simulación deductiva que parece ser el más favorable), crece con una dependencia de G^2 [GOE80], por lo que sería excesivo para circuitos muy complejos. Como ejemplo, si tenemos en cuenta que en la simulación de un circuito de 9478 transistores se invierten del orden de 25 minutos en un ordenador de 12 MIPS, un circuito de 100000 transistores invertirían 47 horas con la dependencia cuadrática [AGR88]. No obstante, prácticamente todos los procedimientos de test precisan de la simulación del circuito, ya sea sin faltas (STAFAN o procedimientos basados en la testabilidad) o con faltas. Esto, que resulta evidente en los procedimientos que pretenden la generación de patrones, también es cierto en los procedimientos pseudoaleatorios y BIST si se desea evaluar la bondad del procedimiento y compararlo con otros. Además, es lógico que ocurra así puesto que habrá que tener en cuenta las características del circuito a testear y sus posibles defectos, dependientes de la tecnología y la forma de realización [ABR86b]. Teniendo esto en cuenta, de lo que se trata es de disminuir el tiempo de simulación obteniendo sin embargo la información suficiente. El enfoque seguido en la presente memoria se basa en dos hipótesis ampliamente aceptadas y utilizadas, aunque presentadas aquí desde otra perspectiva:

(a) Las faltas de anclaje simples permiten modelar gran parte de los defectos físicos en los circuitos integrados (sección 1.2.3).

(b) Se puede realizar la simulación con una muestra de las faltas posibles y aplicar los resultados de la estadística a toda la población [AGR81].

Es decir, se van a considerar muestras de defectos que incluyen sólo a los que se modelan mediante faltas de anclaje simples. Para que esto sea totalmente correcto es preciso que:

(i) Todas las faltas de anclaje modelen adecuadamente algún defecto, tal y como establece la hipótesis (a).

(ii) Los elementos de la muestra se seleccionen de forma aleatoria. Es decir, que si se fuera tomando nota de la secuencia de defectos que se van simulando, no se podría predecir, dado un defecto, cuál es el siguiente en simularse.

De esta manera, el conjunto de los defectos de un circuito integrado, $D=\{d_1, \dots, d_M\}$, se sustituirá en lo que sigue por el conjunto de faltas de anclaje simples que los modelan. No obstante, se sigue utilizando la misma

notación para indicar el conjunto de faltas, es decir d_i será una falta de anclaje simple que modela algún defecto del circuito.

El procedimiento que se presenta se basa en el teorema central del límite, que permite determinar propiedades de la población a partir de los valores de sus estimadores en las muestras tomadas. Dicho teorema permite determinar el número de faltas que han de constituir la muestra; concretamente, si se desea estimar qué valor, p , toma una propiedad de los elementos de una población con una probabilidad de q , y con un error menor que δ , basta tomar una muestra con un conjunto de N elementos según la expresión:

$$N = (Z_r / \delta)^2 p(1-p) \quad (6.23)$$

donde Z_r es una constante tabulada que depende de la probabilidad q , siendo $r=(1-q)/2$. Por ejemplo, en la tabla de Z_r resulta que si se quiere que esa probabilidad sea del 0.99, $r=0.005$ y $Z_{0.005}=2.58$.

Así pues, la obtención de las probabilidades $p(x,e)$, para todos los patrones de error, e , y todas las salidas, x , se lleva a cabo según el siguiente procedimiento:

(1) Seleccionar una muestra Q de N/L faltas de anclaje que se simularán. Para realizar esa selección todas las faltas se consideran equiprobables.

(2) Simular el circuito para cada falta de la muestra, tomando como entradas cada uno de los L patrones de test.

(3.a) Si se utiliza el modelo de n canales binarios independientes: para cada bit de la salida x , x_i ($i=1, \dots, n$), y del patrón de error e , e_i ($i=1, \dots, n$), obtenidos, incrementar en 1 el número $N(e_i/x_i)$, que contabiliza el número de veces que se produce un error e_i en el bit i -ésimo cuando éste es igual a x_i , incrementar en 1 $N(x_i)$, que contabiliza el número de veces que el bit i -ésimo de la salida es x_i , e incrementar en 1 el número de veces $N(x)$ que se tienen a la salida el patrón x .

(3.b) Si se utiliza el modelo de canal q -ario: para cada patrón de salida, x , y patrón de error, e , que se produce, incrementar en 1 el número de veces que debiéndose dar una salida x , se produce la salida xoe , $N(e/x)$.

(4.a) Si se utiliza el modelo de n canales binarios independientes: se obtiene la probabilidad $p'(x,e) = (N(x)/N) (N(e_1/x_1) \dots N(e_n/x_n)) / (N(x_1) \dots N(x_n))$.

(4.b) Si se utiliza el modelo de canal q -ario: se obtiene la probabilidad $p'(x,e) = (1/N) N(e/x)$.

Utilizando (6.23) se verificará que

$$p(x,e) = p'(x,e) + Z_r ((p'(x,e) (1-p'(x,e)) / n)^{1/2}$$



Como se ha podido comprobar, si se utiliza un modelo de canal q-ario, se necesitarían 2^{2n} variables para almacenar los valores de las $N(e/x)$, mientras que, en el caso del canal binario, sólo son necesarias $4n$ variables para las posibles $N(e_i/x_i)$. Sin embargo, con modelos de canales binarios independientes no se pueden modelar adecuadamente situaciones en las que los errores en los bits no sean independientes, cosa que es frecuente en los circuitos VLSI, ya que por su complejidad, normalmente un defecto ocasiona varios bits erróneos, que, por tanto, no serán independientes [PRA90]. Si se presuponen valores de las probabilidades de transición en un modelo u otro se pueden obtener unos valores de las probabilidades $p(x,e)$ cuya adecuación se puede contrastar posteriormente con los resultados experimentales.

Para determinar el tamaño de la muestra para la que se verifique (6.24) con una probabilidad q , añadiendo además la condición de que el error que se cometa sea menor que δ , se puede aplicar (6.25) donde se ha sustituido $p'(x,e)(1-p'(x,e))$ por su valor máximo, es decir 0.25.

$$N=0.25(Z_r/\delta)^2 \quad (6.25)$$

Si por ejemplo, se toma $\delta=0.01$ y $q=0.99$, entonces $Z_{0.005}=2.58$ y $N=16641$.

Dado el test T constituido por los patrones, $y(1), \dots, y(L)$, una vez se han obtenido las probabilidades $p(x,e)$ para todas las parejas de salidas y patrones de error, (x,e) , se puede evaluar la probabilidad de error para el test considerado.

$$p_e = \sum_{x, e \neq 0} p(x, e) \quad (6.26)$$

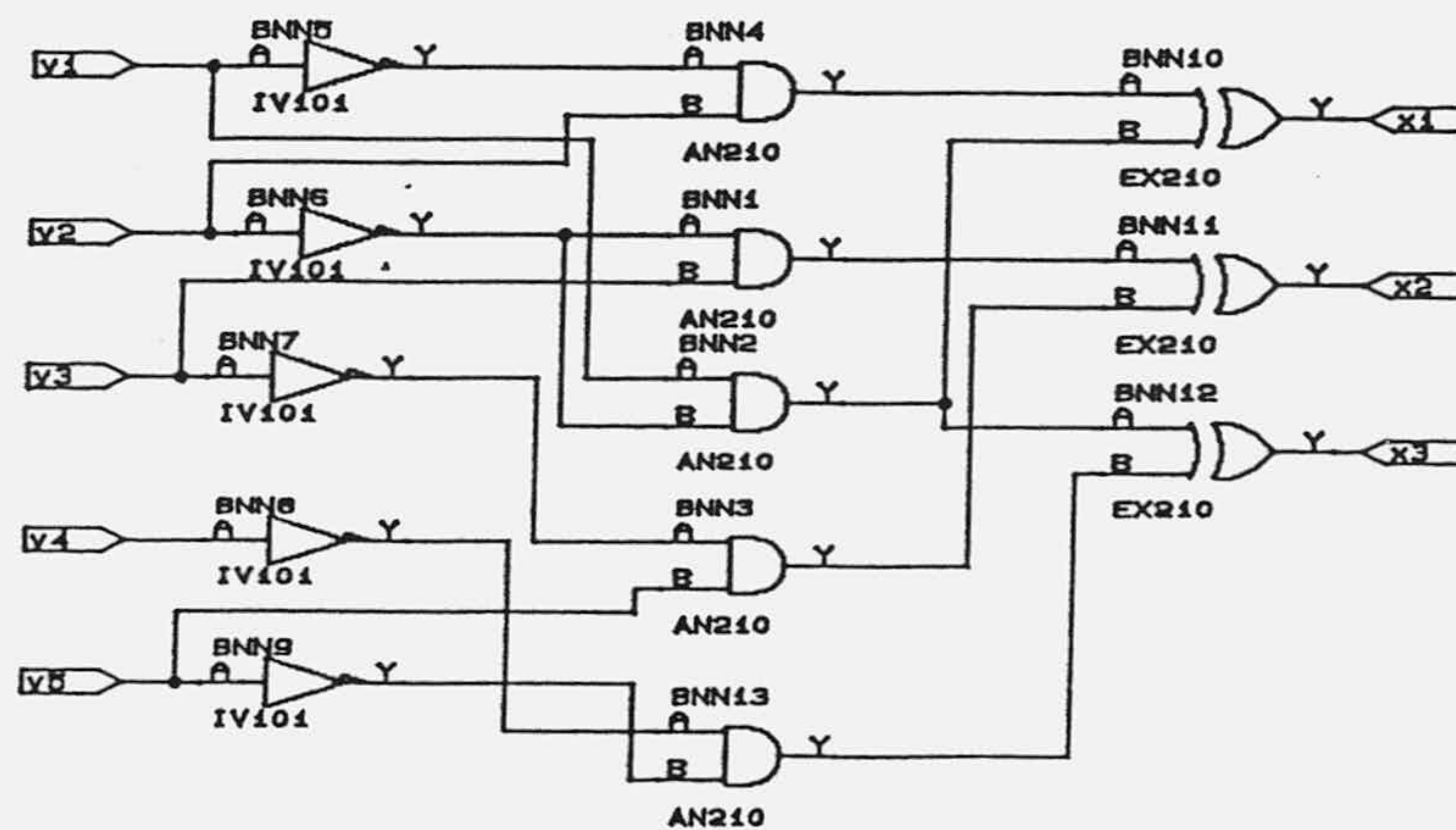


Figura 6.3. Circuito ejemplo para los modelos de errores.

Esta probabilidad es un parámetro que permite comparar conjuntos de patrones de test distintos, ya que cuanto mayor sea la probabilidad de ocasionar errores en las salidas, mayor será la posibilidad de detectar los defectos del circuito. A continuación, las tablas 6.1 y 6.2 muestran los modelos de error obtenidos para el circuito de ejemplo de la figura 6.3. En ambos casos se ha utilizado el mismo conjunto de patrones de test, seleccionado mediante el procedimiento descrito en la sección 3.3, y se han simulado todas las faltas de anclaje simples, debido a que, en este caso el número de faltas de anclaje y el número de patrones de test no es excesivamente elevado. La tabla 6.1 corresponde al modelo de canales de error independientes para cada bit de salida y la tabla 6.2 al modelo de canal q-ario.

Tabla 6.1 Modelo de error con canales binarios.

e	p(e1/x1)		p(e2/x2)		p(e3/x3)	
	x1=0	x1=1	x2=0	x2=1	x3=0	x3=1
0	16/24	1/6	16/21	4/9	8/9	11/21
1	8/24	5/6	5/21	5/9	1/9	10/21

Tabla 6.2 Modelo de error con canal q-ario (q=8).

x\e	100	011	001	101	011
000	-	-	0.034	-	-
100	0.034	-	0.034	0.067	-
010	-	0.034	0.067	0.034	-
001	0.133	0.133	0.067	0.1	0.034
101	0.067	-	-	-	-
011	-	0.133	-	-	0.034

6.4. Procedimiento basado en el Recocido Simulado.

Para la búsqueda de la función de extracción que minimiza la probabilidad de enmascaramiento se ha utilizado como base el algoritmo de Recocido Simulado o Simulated Annealing (RS) [CHA88, JAY88, KIR83]. Este

algoritmo es una técnica de optimización combinatoria de propósito general. Se trata de una extensión del método de Montecarlo [MET53] para determinar el estado de equilibrio de un conjunto de partículas a una temperatura dada. Su nombre se debe a la analogía que existe entre la solución de problemas de optimización combinatoria y el proceso de recocido, en el que un material es calentado y enfriado muy lentamente.

En un material muy caliente sus partículas se pueden mover aleatoriamente y, en este estado de alta energía, las probabilidades de que los átomos se muevan en cualquier dirección son idénticas. Cuando la sustancia se enfría lentamente, las partículas pierden movilidad y tienden a organizarse en una estructura cristalina que corresponde al estado de mínima energía del sistema. En cambio, si el enfriamiento se produce muy rápidamente se obtiene un estado policristalino que corresponde a un mínimo local de la energía del sistema. Cuando se utiliza el RS en un problema de optimización combinatoria, debe establecerse una función que permite estimar el costo de la solución que se ha obtenido y que, en el caso termodinámico corresponde a la energía. En la aplicación que se considera aquí se utilizará la probabilidad de enmascaramiento dada en (6.17).

Así pues, el problema a resolver se puede expresar en los siguientes términos: seleccionar de entre un total de r coeficientes espectrales, α_i ($i=1, \dots, r=2^n-1$), un número no superior a P de manera que se minimize la figura de mérito representada por la probabilidad de enmascaramiento p_A . Expresado de esta forma queda de manifiesto su equivalencia con el problema de la asignación estática de r tareas a P procesadores minimizando una función de costo C . Pues bien, éste es un problema NP-Completo [MAY89] y por consiguiente el problema que se plantea aquí también lo es.

En el algoritmo que se presenta a continuación (Algoritmo 6.4), basado en el RS, S_α es el conjunto de coeficientes espectrales que se van seleccionando y que finalmente constituirán la solución buscada definiendo la función de extracción. La probabilidad de enmascaramiento, p_A , se calcula a partir de los coeficientes incluidos en S_α según (6.17). La temperatura T es un parámetro cuyo valor determina la probabilidad con la que se acepta un cambio que aumenta la energía del sistema. Esta probabilidad viene dada por la función de Boltzmann $e^{-\delta P_A/T}$. A medida que el algoritmo progresa, la probabilidad de aceptar una configuración que empeora el valor de p_A debe disminuir, por lo que el valor de la temperatura debe ir haciéndose más pequeño. Si la temperatura se fija a un valor infinito, la probabilidad de aceptar una solución peor es 1 y el algoritmo se convierte en una prueba

aleatoria de todas las soluciones posibles, y si la temperatura se fija a cero, el algoritmo continúa hasta que encuentra un primer mínimo local, donde queda atrapado. Por otra parte, una vez fijada una temperatura, los cambios que empeoran la probabilidad de enmascaramiento, p_A , en menor cuantía tienen una mayor probabilidad de aceptarse.

El algoritmo se inicia con la selección al azar de una configuración, caracterizada por una temperatura T , un conjunto de coeficientes espectrales S_α , y la probabilidad de enmascaramiento p_A correspondiente. A partir de cada configuración seleccionada se generan nuevas configuraciones. Si una de esas nuevas configuraciones tiene asociada una probabilidad de enmascaramiento p_A menor que la seleccionada previamente, se toma como nueva configuración. El proceso continúa hasta que se haya seleccionado una configuración que verifique la condición de fin o la condición de equilibrio. El algoritmo es:

Algoritmo 6.4:

1. Generar una configuración inicial: T, S_α, p_A .
2. Si se ha alcanzado la Condición de Fin (CF), terminar.
3. Si se ha alcanzado la Condición de Equilibrio (CE), terminar.
4. Seleccionar una nueva configuración con menos de P elementos: S_α (nueva)
5. Evaluar el costo para la nueva configuración, p_A (nueva), y calcular $\delta p_A = p_A$ (nueva) - p_A .
6. Si $\delta p_A \leq 0$ ir a 8.
7. Si $\delta p_A > 0$ se selecciona aleatoriamente si se va a 8 o a 9. La probabilidad de ir a 8 es $e^{-\delta p_A/T}$ y, por consiguiente la de ir a 9 es $1 - e^{-\delta p_A/T}$.
8. Hacer $S_\alpha = S_\alpha$ (nuevo) y $p_A = p_A$ (nuevo).
9. Actualizar la temperatura T .
10. Ir a 2. ■

Para poder concretar la programación del algoritmo 6.4 es preciso considerar las siguientes cuestiones:

- (a) La forma de seleccionar nuevas configuraciones.
- (b) La función mediante la que se actualiza la temperatura, T .
- (c) La evaluación de la condición de fin (CF).
- (d) La evaluación de la condición de equilibrio (CE).

Las decisiones que se han adoptado respecto a ellas son las siguientes:
 - La selección de la nueva configuración se hace aleatoriamente, manteniendo el número de coeficientes distintos de cero en la nueva configuración, siempre por debajo de P .

- La temperatura se va actualizando según una función del tipo $T(\text{nueva}) = \beta T$, con $\beta < 1$. Es decir, se va disminuyendo lentamente. Para obtener el valor de β se tiene en cuenta la ley de actualización de la temperatura $T_j = \beta T_{j-1}$. Así, llamando T_0 a la temperatura inicial, se tiene que $T_j = \beta^j T_0$. Por otra parte, puesto que el número posible de coeficientes para un circuito con n salidas es 2^n , y la probabilidad de tomar una configuración dada que empeora la función de costo es en el peor de los casos $e^{-1/T}$, cuando la temperatura es T , se puede imponer la condición de que la probabilidad de seleccionar las 2^n primeras configuraciones sea lo suficientemente elevada de manera que se explore el espacio de posibles soluciones. Así, se impone la condición de que $\exp(-1/T_j) > 0.99$ para $j < b2^n$, donde b es una constante de proporcionalidad para fijar el número de configuraciones que se considere necesario seleccionar antes de que la probabilidad sea menor que 0.99. Es decir:

$$\ln(999.5/T_0) < \ln(\beta)N \quad (N=b2^n) \quad (6.27)$$

Mediante (6.27), y teniendo en cuenta el número de coeficientes espectrales que definen la configuración en el problema a resolver, 2^n , y el parámetro b , se puede determinar el menor valor de β aceptable una vez establecida la temperatura inicial, T_0 , o bien fijar ésta una vez dada β . En la figura 6.4 se muestran las curvas que relacionan los valores de β y N para algunos valores de la temperatura inicial.

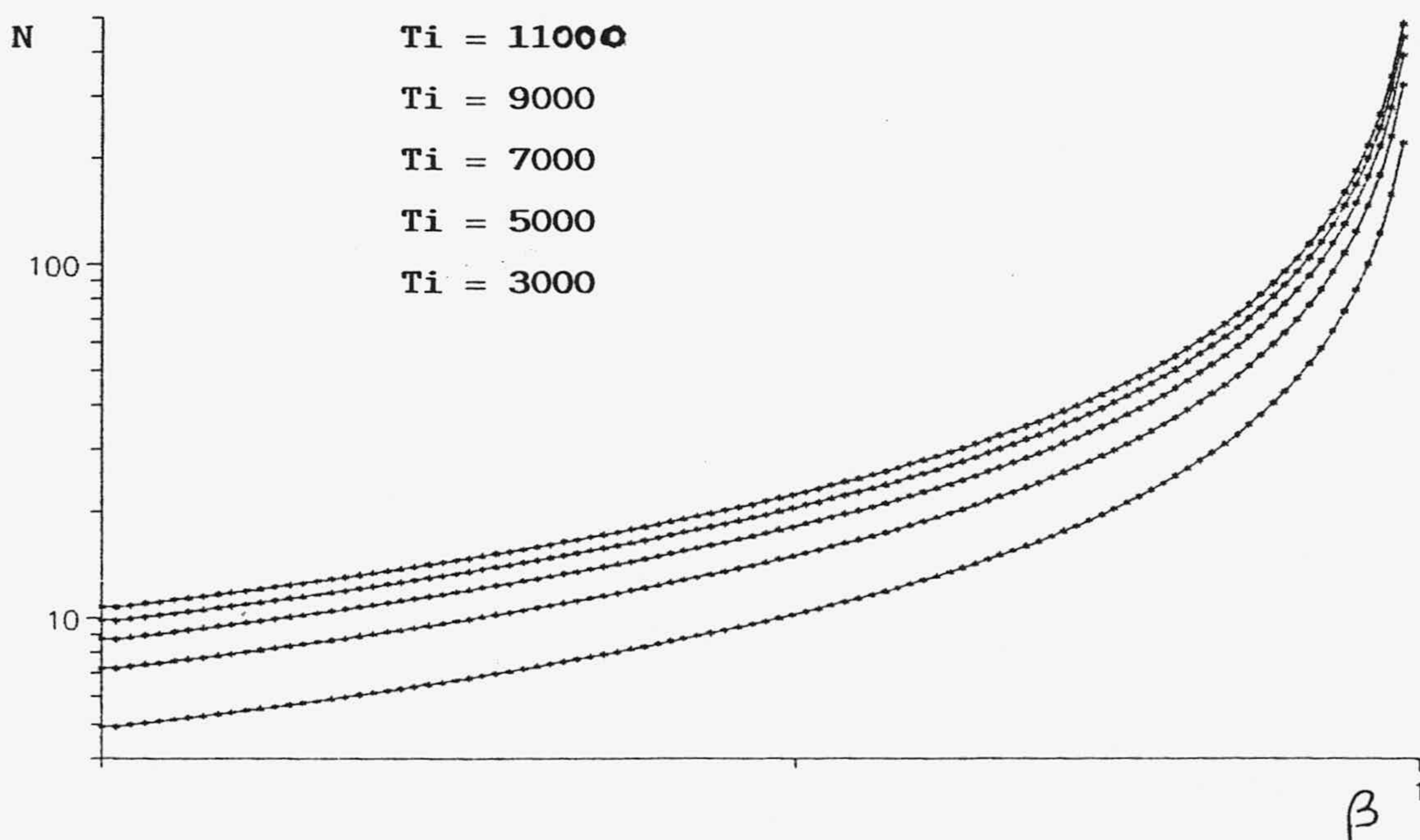


Figura 6.4. Curvas de N frente a β para distintas T_0 .

- La condición de fin (CF) se produce al llegar a una temperatura suficientemente baja como para que la probabilidad de aceptar una solución peor sea despreciable. Por ejemplo, si se toma $\exp(-1/T_f)=0.0001$, se tiene que $T_f=-1/\ln(0.0001)$ es la condición de fin.

- La condición de equilibrio (CE) se produce cuando se ha dado un número suficientemente elevado de soluciones aceptadas (SA). Se ha demostrado [OTT88] que según aumenta el número de SA para que se considere alcanzada la CE, las soluciones mejoran pero sólo hasta un cierto valor de SA. La condición de equilibrio se establece teniendo en cuenta que, puesto que 2^n es el número de coeficientes espectrales posibles, si el número de soluciones aceptadas es $b2^n$ se puede considerar que se ha alcanzado la condición de equilibrio.

Como a temperaturas bajas la condición de SA es difícil de verificar, para obtener CE a dichas temperaturas se utiliza la condición de que se hayan generado S soluciones.

La Tabla 6.3 muestra los resultados obtenidos por el algoritmo presentado para diferentes modelos de error, obtenidos considerando fuentes de error binarias independientes en cada uno de los n bits de los patrones de salida. Al aplicar el algoritmo al modelo de errores de la Tabla 6.2 para el circuito de la figura 6.3, con valores de $\beta=0.99$, $T_0=3000$, se obtuvo como función de extracción $g_1(x_1, x_2, x_3)=x_2 \oplus x_3 \oplus x_1 x_2$ y una probabilidad de enmascaramiento igual a $p_{A1}=0.133$. Con esta función de extracción no se detectaría error para las parejas de salida y patrón de error, $(x,e)=(100,100)$, $(100,001)$, $(010,101)$, $(001,011)$, $(011,011)$. Si la probabilidad de enmascaramiento obtenida no fuera aceptable, se volvería a aplicar el algoritmo 6.4 considerando únicamente los patrones de error no detectados. En ese caso se obtiene $g_2(x_1, x_2, x_3)=x_1 \oplus x_2 x_3 \oplus x_3$ y la probabilidad de enmascaramiento $p_{A2}=0$. Consiguientemente, con las funciones de extracción g_1 y g_2 se detectarían todos los posibles patrones de errores que se pueden producir cuando se aplican los patrones del test seleccionado y el circuito se ve afectado por un defecto que se modela mediante una falta de anclaje simple en alguno de sus nudos.

6.5. Conclusión.

En este capítulo se ha elaborado una medida para evaluar el módulo extractor en un circuito testeable concurrentemente. Además, se ha puesto de manifiesto que el problema de seleccionar la función de extracción óptima es

NP-Completo y se ha presentado un algoritmo para la determinación de dicho módulo de extracción. En el Apéndice V se consideran las posibilidades que ofrecen las redes neuronales en este problema de optimización utilizando una red neuronal de Hopfield generalizada para determinar la función de extracción.

Una vez considerada la determinación de las funciones de extracción, en el siguiente capítulo se estudian los aspectos relativos a la síntesis del módulo predictor y monitor de patrones. Con esto se habrán analizado todos los módulos a añadir a un CUT para hacerlo testeable concurrentemente.

Tabla 6.3. Aplicación del algoritmo RS a algunos modelos de errores (con fuentes de error binarias independientes).

n	$p_1(1/0), p_1(1/1); p_2(1/0), p_2(1/1); p_3(1/0), p_3(1/1); p_4(1/0), p_4(1/1); p_5(1/0), p_5(1/1)$ $g_1, p_{A1}; \dots; g_r(x_1 x_2 x_3 x_4 x_5), p_{Ar}$	$T_1; \beta; N$ p_{AT}
3	0.1, 0.1; 1.0, 1.0; 0.2, 0.2 $g_1=x_2, p_{A1}=0$	80;3000;.986 $p_{AT}=0$
3	1.0, 1.0; 0.1, 0.1; 1.0, 1.0 $g_1=x_1, p_{A1}=0$	80;3000;.986 $p_{AT}=0$
3	0.2, 0.2; 0.8, 0.8; 0.1, 0.1 $g_1=x_2, p_{A1}=0.65; g_2=x_1 \otimes x_3, 0.0071$	800;3000;.998 $p_{AT}=.0046$
3	0.1, 0.1; 0.85, 0.85; 0.8, 0.8 $g_1=x_2, p_{A1}=0.129; g_2=x_3, 0.0477$	800;3000;.998 $p_{AT}=.006$
3	0.1, 0.75; 0.1, 0.1; 0.1, 0.1 $g_1=x_1 \otimes x_2 \otimes x_3, 0.154; g_2=x_1 \otimes x_2, 0.07$	800;3000;.998 $p_{AT}=.0107$
4	0.2, 0.2; 0.8, 0.8; 0.1, 0.1; 0.1, 0.1 $g_1=x_2, 0.081; g_2=x_1 \otimes x_3 \otimes x_4, 0.125$	160;3000;.993 $p_{AT}=.0101$
4	0.1, 0.1; 0.85, 0.85; 0.8, 0.8; 0.85; 0.85 $g_1=x_4, 0.147; g_2=x_2, 0.127$	1600;5000;.998 $p_{AT}=.019$
4	0.1, 0.1; 0.1, 0.1; 0.1, 0.1; 0.1, 0.1; 0.1, 0.1 $g_1=x_1 \otimes x_2 \otimes x_3 \otimes x_4, 0.142; g_2=x_1 \otimes x_4, 0.33$	1600;5000;.998 $p_{AT}=.047$
4	0.0, 0.1; 0.0, 0.95; 0.0, 0.1; 0.0, 0.1 $g_1=x_1 \otimes x_2 \otimes x_3 \otimes x_4, 0.124; g_2=x_2, 0.055$	1600;5000;.998 $p_{AT}=.007$
5	0.2, 0.2; 0.8, 0.8; 0.1, 0.1; 0.1, 0.1; 0.2, 0.2 $g_1=x_2, 0.107; g_2=x_1 \otimes x_3 \otimes x_4 \otimes x_5, 0.2$	320;5000;.995 $p_{AT}=.021$
5	0.1, 0.1; 0.1, 0.1; 0.1, 0.1; 0.1, 0.1; 0.1, 0.1 $g_1=x_1 \otimes x_2 \otimes x_3 \otimes x_4 \otimes x_5, 0.1808; g_2=x_2 \otimes x_5, 0.402$	3200;5000;.9995 $p_{AT}=.073$
5	0.2, 0.0; 0.1, 0.0; 0.9, 0.0; 0.3, 0.0; 0.1, 0.0 $g_1=x_2 \otimes x_3 \otimes x_4 \otimes x_5 \otimes x_1 \otimes x_3 \otimes x_2, 0.239; g_2=x_1 \otimes x_2 \otimes x_4 \otimes x_5, 0.124$	3200;5000;.9995 $p_{AT}=.003$

CAPITULO 7. DISEÑO DE LOS MODULOS PREDICTOR Y MONITOR DE PATRONES.

En el capítulo 5 se describió el procedimiento para sintetizar los módulos que permiten hacer un circuito testeable concurrentemente. En ese procedimiento se distinguen dos partes: la síntesis del módulo extractor y la síntesis del módulo predictor y monitor de patrones. En el capítulo 6 se llevó a cabo el estudio de los problemas que plantea la síntesis del módulo extractor y se describieron varios procedimientos para realizarla. El presente capítulo se dedica a los aspectos relativos a la síntesis del módulo predictor y monitor de patrones (figura 5.7), y las cuestiones a considerar en su diseño se refieren a la selección del conjunto de patrones de test para el que:

- (a) es máxima la probabilidad p_e de producir un error en las salidas cuando se aplica una entrada y el circuito es defectuoso, y
- (b) es mínima la complejidad de los circuitos que realizan el módulo.

El valor de p_e para un conjunto de patrones de test viene dado por:

$$p_e = \sum_{Y \in T, E \neq 0} p(E/Y) p(Y) \quad (7.1)$$

Así pues, una forma de incrementar p_e consiste en aumentar el número de patrones de test. Ahora bien, cuanto mayor sea ese número, menos indiferencias tendrá la función a realizar por el módulo de predicción y, en general, aumentará la complejidad del circuito. Por consiguiente, hay que establecer un compromiso entre la tendencia a disminuir el número de patrones para reducir la complejidad del circuito y la tendencia a aumentarlo para tener una

probabilidad p_e mayor y reducir así el tiempo de latencia. A esta cuestión se dedica el presente capítulo.

En la sección 7.1 se presentan los aspectos a tener en cuenta para seleccionar los patrones de test, centrándose fundamentalmente en los procedimientos aleatorios de selección. La sección 7.2 se dedica a la explicación detallada del procedimiento de selección aleatoria que proponemos, en tanto que la sección 7.3 presenta varios procedimientos de selección basados en la transformada de Tamari. En la sección 7.4 se considera la síntesis del módulo predictor y del módulo monitor de patrones. Finalmente, la sección 7.5 es la conclusión del capítulo.

7.1. Selección de los patrones de test.

En esta sección se estudia el problema de la selección del conjunto de patrones de test que permita una síntesis razonablemente simple de los módulos de monitorización de patrones y de predicción y para los que la probabilidad de ocasionar errores en las salidas sea suficientemente elevada.

Considerese un circuito R^F que sintetiza N funciones de m variables. Las N funciones se notarán con $F=(f_1, f_2, \dots, f_N)$ y las entradas con $Y=(Y_0, Y_1, \dots, Y_{m-1})$. La salida del circuito para la entrada Y se expresa mediante $F(Y)=(f_1(Y), \dots, f_N(Y))$.

La probabilidad p_e de que un conjunto T de combinaciones de entradas ocasione un error en las salidas de un circuito por cualquiera de los defectos posibles, se obtiene a partir de (7.1) que, si se hace

$$p_e(Y) = p(Y) \{ \sum_{E \neq 0} p(E/Y) \} \quad (7.2)$$

se puede escribir como

$$p_e = \sum_Y p_e(Y) \Theta_T(Y) \quad (7.3)$$

donde $p_e(Y)$ es la probabilidad de que se ocasione un error en las salidas para cualquier defecto al aplicar el patrón Y , la sumatoria se extiende a todas las posibles combinaciones de entradas, $Y \in F_2^m$ para un circuito con m entradas, y $\Theta_T(Y)$ es una función que ha de verificar:

$$\Theta_T(Y) = \begin{cases} 1 & \text{si } Y \in T \\ 0 & \text{en caso contrario.} \end{cases} \quad (7.4)$$

La probabilidad p_e de que una combinación de entradas, $Y \in F_2^m$, origine un error a la salida del circuito cuando éste se ve afectado por un defecto cualquiera se llama efectividad de la entrada Y . La probabilidad p_e de que se

origine un error cuando se aplica alguno de los patrones de un test es la efectividad del test T . Así pues, mientras que la detectabilidad de una falta indica cuántos patrones de test pueden ponerla de manifiesto [WAG87], la efectividad de una entrada da información del número de faltas que dicha entrada puede detectar.

Seleccionar un conjunto de patrones de test es equivalente a determinar la función Θ_T (7.4) y, por consiguiente, esta es la función que ha de sintetizar el módulo monitor de patrones, ya que su cometido es indicar cuándo la combinación de entradas aplicada al circuito es un patrón de test (salida 1, por ejemplo) o no lo es (salida 0).

Expresando Θ_T en términos de los coeficientes espectrales de Tamari,

$$\Theta_T(Y) = \sum_j \alpha_j T_j(Y) \quad (7.5)$$

(7.3) se transforma en

$$p_e = \sum_Y p_e(Y) [\sum_j \alpha_j T_j(Y)] \quad (7.6)$$

donde $j=0,1,\dots,2^m-1$ y $T_j(Y)$ es la función j -ésima de la base de Tamari en F_2^m . Teniendo en cuenta (6.14) y definiendo los conjuntos

$$S(Y) = \{j \in C_2^m / j_k \leq Y_k \quad \forall k=0,1,\dots,m-1\} \quad (7.7)$$

(7.6) se puede expresar como:

$$p_e = \sum_Y p_e(Y) [\sum_{j \in S(Y)} \alpha_j] \quad (7.8)$$

Si se utiliza la probabilidad de no ocasionar un error en las salidas cuando se aplica una entrada, p_{ne} , puesto que $p_{ne} = 1 - p_e$, utilizando (7.8) se tiene

$$p_{ne} = 1 - \sum_Y p_e(Y) [\sum_{j \in S(Y)} \alpha_j] \quad (7.9)$$

y maximizar (7.8) implica, por consiguiente, minimizar (7.9).

Esta expresión tiene la misma forma que la (6.17) para evaluar la probabilidad de enmascaramiento del módulo extractor. De esta manera, los mismos algoritmos que se utilizaron para obtener el mínimo de (6.17) se podrían utilizar para minimizar (7.9) y determinar la función a realizar por el monitor de patrones. No obstante, hay que tener en cuenta que en este caso concreto, además de minimizar (7.9), interesa:

(a) que el número de patrones de test, es decir, el número de elementos distintos de cero de Θ_T sea el menor posible, y

(b) que la complejidad del monitor de patrones sea la menor posible, con lo que el número de coeficientes de Tamari no nulos debe minimizarse también.

Además, en (7.8) aparecen las probabilidades $p_e(Y)$. Puesto que la sumatoria se extiende a todas las posibles combinaciones de entrada, es necesario obtener los valores de las $p_e(Y)$ para todas ellas a fin de que quede perfectamente determinada (7.8). Esto se puede hacer a partir de (7.2), donde los valores de $p(E/Y)$ se pueden estimar a partir de un muestreo de faltas, según se vió en la sección 6.3 para obtener las distribuciones de error. Hay que tener en cuenta también que deben obtenerse los $p_e(Y)$ para todos los $Y \in F_2^m$. Si el número de entradas al circuito R^F es moderadamente grande, la cantidad de combinaciones que hay que considerar puede ser excesivamente elevada. Sería preciso determinar un conjunto previo de patrones de entradas; aunque esto conlleva eliminar de partida posibles patrones de test, existe un criterio claro para seleccionar ese conjunto de patrones. Concretamente, para cada uno de los defectos posibles del circuito (normalmente para cada una de las faltas de anclaje simples de los nudos del circuito) debe existir en T al menos un patrón de test que lo detecte, ya que en caso contrario, si se produjese ese defecto no se advertiría. Por ello, el conjunto previo de patrones de entradas debe garantizar un 100% de cubrimiento de faltas. Además, también hay que tener en cuenta que el circuito de extracción enmascara algunos de los errores que ocasionan los defectos cuando se aplican al circuito ciertos patrones y, por tanto, el test también debe ser capaz de detectar todos los defectos del circuito si se observasen únicamente las salidas del extractor. Así pues, el proceso constaría de estos tres pasos:

1. **Seleccionar patrones de test para el circuito que permitan un 100% de cubrimiento de faltas.**
2. **Determinar el circuito extractor para ese conjunto de test.**
3. **Comprobar si se mantiene el 100% de cubrimiento de faltas cuando se observan sólo las salidas del extractor. En caso contrario se buscan los patrones que completan el cubrimiento.**

Aunque (7.8) no se puede utilizar para la síntesis del monitor de patrones si no se conocen las efectividades de todas las combinaciones de entradas, sí que permite la evaluación de la efectividad de un test, T . Para eso, se estiman las efectividades de los patrones del test mediante muestreo de faltas (sección 6.3) y simulación para cada uno de los patrones del test, $Y \in F_2^m$. Como se conoce la función Θ_T , sustituyendo se obtiene la efectividad del test. Conocida p_e y la probabilidad de enmascaramiento del módulo extractor, se puede determinar el tiempo de latencia del método.

Así pues, en esta sección se han determinado las expresiones que permiten evaluar la efectividad de un conjunto de patrones de test y se han mostrado los problemas que plantea su utilización en la síntesis del módulo predictor. En los siguientes apartados se considerarán los procedimientos que se han utilizado para la selección de los patrones.

En el capítulo 1 (sección 1.2.2) ya se consideró el problema de la selección de patrones y se distinguieron dos tipos esenciales de procedimientos: los procedimientos deterministas y los procedimientos aleatorios o pseudoaleatorios. Cada uno de ellos corresponde a un planteamiento totalmente diferente y lleva aparejados problemas distintos.

7.1.1. Selección determinista.

Estos procedimientos se han considerado en los capítulos 1, 2, 3 y 4, y pretenden definir un conjunto mínimo de combinaciones de entradas para poner de manifiesto cualquier posible defecto del circuito que ocasione un funcionamiento erróneo. Es preciso conocer la forma en la que cada defecto afecta a las salidas del circuito al aplicar las distintas entradas. Por otra parte, los defectos que puede presentar cada circuito dependen de la tecnología de fabricación.

En la práctica se realiza un modelado de los defectos a un nivel de descripción manejable para los cálculos y las simulaciones, normalmente el nivel de puertas lógicas o el de transistor como conmutador. Dada una falta que se supone modela adecuadamente algún defecto del circuito, los procedimientos deterministas permiten obtener las entradas que la ponen de manifiesto a la salida. Esto nos sitúa ante otro problema: el número de defectos que puedan afectar al circuito que se testea.

Si se considera el nivel de representación de puertas lógicas y aceptamos como válida la hipótesis de que existe una aplicación biunívoca entre defectos físicos y faltas de anclaje, el número de defectos posibles en un circuito (faltas de anclaje posibles) es una función exponencial del número de puertas. Aún para circuitos razonablemente simples, el número de posibles defectos es extraordinariamente elevado. En la práctica sólo se consideran las faltas de anclaje simples, cuyo número es proporcional al de puertas. En circuitos VLSI, ese número también puede ser demasiado elevado.

Una posible solución es considerar un conjunto de patrones de test elegidos de manera aleatoria. Mediante simulación se determina el conjunto de

faltas de anclaje simples que permiten detectar y se aplican los procedimientos deterministas para encontrar los patrones de test que detecten el resto de faltas de anclaje simples. De esta forma, se plantea el tema de la simulación de faltas, que se estudiará más adelante, cuando se hayan considerado los procedimientos de generación no deterministas, para cuya evaluación se utilizan simuladores de faltas.

Así pues, de cara a la síntesis del módulo monitor de patrones, los métodos deterministas plantean los siguientes problemas:

- (a) Se necesita un modelo adecuado de los defectos.
- (b) Si el número de faltas es elevado, el tiempo necesario para generar los patrones es excesivo.
- (c) El circuito monitorizador de patrones puede ser demasiado complejo.

El costo del proceso de generación determinista de patrones de test para un circuito de G puertas crece, al menos, proporcionalmente a G^2 [GOE80]. Por consiguiente en el caso de circuitos con un número elevado de puertas puede resultar desaconsejable este procedimiento.

7.1.2. Selección aleatoria.

Una forma de evitar la búsqueda de un patrón para cada uno de los defectos posibles consiste en seleccionar aleatoriamente un conjunto de patrones lo suficientemente grande y estudiar si ofrecen una cobertura de faltas suficiente. En este caso se habla de procedimientos de selección aleatorios o pseudoaleatorios. Esta sección se dedica a su estudio.

Aunque los procedimientos pseudoaleatorios de selección de patrones permiten eludir la búsqueda de patrones para una falta dada, plantean el de la evaluación de su efectividad: cómo determinar el número de patrones de test a seleccionar para tener un cubrimiento de faltas dado. Una vez seleccionado el conjunto de patrones se puede utilizar (7.8) para evaluar su efectividad. Si ésta no es suficiente se puede aumentar el número de patrones hasta que se alcancen los valores deseados. Sin embargo este proceso implica sucesivas simulaciones de los patrones de test obtenidos y puede ser excesivamente costoso en lo que a tiempo se refiere. Por esa razón se han elaborado medidas para evaluar la cobertura de un conjunto de patrones seleccionados aleatoriamente y para obtener el número de patrones a seleccionar si se quiere asegurar una cobertura mínima establecida [MAL84, SAV84, WAG87, MCL88].

La cuestión que hay que resolver cuando se utiliza un procedimiento de test aleatorio es la de estimar el cubrimiento de defectos que se consigue para un conjunto determinado de vectores de test seleccionado aleatoriamente, o bien, el problema inverso, es decir, determinar el número de vectores de test elegidos aleatoriamente con los que se puede conseguir un cubrimiento de defectos dado. También aquí se necesita medir el cubrimiento de defectos para evaluar si los patrones de test que se han seleccionado son lo suficientemente buenos. Por tanto, se pueden repetir todo lo dicho al respecto en la sección anterior y, consecuentemente, habrá que utilizar faltas y estimar el cubrimiento de defectos a partir del cubrimiento de faltas.

A continuación, se obtienen las expresiones que permiten evaluar o diseñar, según se mire, un procedimiento de test aleatorio [MAL84, WAG87].

Sean $p(k,i)$, la probabilidad de que el patrón de test que se aplica en el k -ésimo lugar sea el primero que detecte el defecto modelado con la falta f_i ; M , el número de defectos que puede presentar el circuito; y $d(k,i)$, una función tal, que $d(k,i)=1$ si el defecto modelado por la falta f_i es detectado por primera vez por el patrón aplicado en k -ésimo lugar, y $d(k,i)=0$ en los demás casos. El número de defectos detectados por primera vez por el patrón k -ésimo viene dado por:

$$N(k) = \sum_{i=1}^M d(k,i) \quad (7.10)$$

y su valor medio será:

$$N(k) = \sum_{i=1}^M p(k,i) \quad (7.11)$$

Así, para calcular el cubrimiento de defectos esperado que se tiene al aplicar k patrones, se divide el número medio de defectos detectados, $N(k)$, por el número de defectos:

$$C(k) = (1/M) (\sum_{i=1}^M p(k,i)) \quad (7.12)$$

Ahora bien, para poder evaluar $C(k)$ se ha de disponer de los valores de las probabilidades $p(k,i)$. Es lo que se aborda a continuación.

Si para un número de combinaciones de entradas n_i de las N aplicables a un circuito, el defecto modelado por la falta f_i ocasiona una salida errónea, la probabilidad de que una entrada elegida al azar detecte el defecto es n_i/N . Consecuentemente, la probabilidad de que f_i no se detecte hasta que se aplique el patrón k -ésimo viene dada por:

$$p(k,i) = (n_i/N) (1 - n_i/N)^{k-1} \quad (7.13)$$



No obstante, (7.13) es válida sólo si una vez que se haya aplicado un patrón de test, se puede volver a aplicar. Es la situación que se da en el test concurrente, puesto que cada entrada se puede presentar en cualquier momento.

En el caso de que, una vez aplicado un patrón de test, este no se vuelva a aplicar más, la probabilidad de que el patrón j -ésimo detecte la falta f_i si ninguno de los $j-1$ patrones anteriores la detectó, viene dada por $X(j,i)=n_i/(N-j+1)$, ya que cada vez que se aplica un patrón queda una combinación de entradas menos. Así pues:

$$p(k,i)=[\prod_{j=1}^{k-1} (1-X(j,i))]X(k,i)=n_i [(N-n_i)!(N-k)!]/[N!(N-n_i-k+1)!] \quad (7.14)$$

$(j < k < n)$

De esta manera, sustituyendo en (7.12) $p(k,i)$ dada por (7.13) o por (7.14), según el caso, se dispone de una expresión para calcular el cubrimiento de defectos esperado en un procedimiento de test aleatorio. Como se puede ver, esa expresión depende de la longitud del test, del número de entradas, N ; del número de defectos, M ; y del número de entradas que permiten detectar cada falta, n_i .

La expresión (7.12) se puede modificar tal como sigue:

$$C(k) = (\sum_{i=1}^M \sum_{n_i=1}^N \Omega(n_i) p(k,i)) / M \quad (7.15)$$

donde aparece el número de defectos que pueden ser detectados por n_i patrones de test, $\Omega(n_i)$, y puesto que todos ellos tienen los mismos valores de $p(k,i)$, se verifica que

$$\sum_{n_i=1}^N \Omega(n_i) p(k,i) = \sum_{i=1}^M p(k,i) \quad (7.16)$$

El vector $\Omega = (\Omega(1), \dots, \Omega(N))$ es el vector de detectabilidades [MAL84, WAG87]. Si se sustituye (7.13) ó (7.14) en (7.15) se obtiene una expresión del cubrimiento esperado en función de k , N , M , n_i y $\Omega(n_i)$.

Para determinar la longitud del test que asegura un cubrimiento de faltas dado se parte de la probabilidad de que la falta i -ésima sea detectada por un conjunto de L patrones seleccionados aleatoriamente, $c_L(i)$. El valor de $c_L(i)$ se obtiene a partir de

$$c_L(i) = \sum_{k=1}^L p(k,i) \quad (7.17)$$

y si $p(k,i)$ se sustituye por (7.13) ó (7.14), se puede expresar (7.15) en términos de la detectabilidad de la falta i -ésima considerada. Así, si se sustituye (7.13) en (7.15), se obtiene

$$c_L(i) = \sum_{k=1}^L (n_1/N) (1-n_1/N)^{k-1} = 1 - (1 - (n_1/N)^L) \quad (7.18)$$

de donde

$$(1 - c_L(i)) = (1 - (n_1/N)^L) \quad (7.19)$$

y para L queda

$$L = \log(1 - c_L(i)) / \log(1 - (n_1/N)) \quad (7.20)$$

Si se sustituye (7.14) en (7.15), se obtiene

$$\begin{aligned} c_L(i) &= \sum_{k=1}^L n_1 [(N-n_1)! (N-k)!] / [N! (N-n_1-k+1)!] = \\ &= 1 - \{ [(N-L)! (N-n_1)!] / [N! (N-n_1-L)!] \} \quad (N-L) > n_1 \end{aligned} \quad (7.21)$$

y de (7.21) resulta

$$(1 - c_L(i)) [N! (N-n_1-L)!] = [(N-L)! (N-n_1)!] \quad (7.22)$$

que, si se asume que $(N-L) \gg n_1$, se transforma en

$$(N-L)^{n_1} = N^{n_1} (1 - c_L(i)) \quad (7.23)$$

de donde

$$L = N(1 - (1 - c_L(i))^{1/n_1}) \quad (7.24)$$

Así pues, las expresiones (7.20) y (7.24) permiten determinar, en cada caso, el número de patrones de test que hay que utilizar para asegurar que la falta i -ésima, que modifica las salidas en n_1 combinaciones de entradas, se detectará con una probabilidad $c_L(i)$. L crece según n_1 disminuye, lo que es lógico, ya que cuanto menor sea el número de combinaciones de entradas que ponen de manifiesto una falta, mayor será el número de patrones que hay que seleccionar para asegurar que alguno de ellos detecte la falta. Según esto, para determinar el número L de patrones de test que hay que aplicar basta utilizar en (7.20) o en (7.24) el valor mínimo que toma n_1 una vez fijado un valor de $C_L(i)$ mínimo para todas las faltas.

El valor de k es un parámetro de procedimiento de test, y N se obtiene directamente a partir del número de entradas n del circuito ($N=2^n$). Sin embargo, los valores del número de defectos M , del número de entradas que detectan cada defecto, n_1 , y de las detectabilidades, $Q(n_1)$, no son tan fáciles de obtener. Para su determinación hay que recurrir a modelar los defectos. Así, se contabiliza el número de defectos como número de faltas

simples y el vector Ω y los n_i se obtienen por simulación o bien mediante métodos estadísticos como el STAFAN (sección 1.2.4) [JAI84, 85].

La determinación del valor mínimo de n_i necesario en (7.20) y (7.24) se puede hacer de forma directa si se conoce el vector de detectabilidades. Si no se conoce, es posible determinar n_i mediante el análisis de testeabilidad del circuito y simulación de las faltas con menor testeabilidad. El costo de simulación es uno de los aspectos que hay que tener en cuenta al seleccionar un procedimiento de generación de patrones de test, aunque la importancia de un procedimiento de selección aleatoria radica en la posibilidad de su implementación en hardware, lo que lo hace apropiado para el test BIST.

La complejidad de un simulador de faltas en términos de tiempo de CPU y necesidad de memoria es proporcional al cuadrado del número de puertas del circuito [GOE82]. En el caso de circuitos VLSI esa dependencia representa un serio obstáculo. Una forma de evitar la simulación es la técnica de **Testeabilidad de Patrones Aleatorios (Random Pattern Testability)** basada en el **Algoritmo de Corte (Cutting Algorithm)**, [SAV83] y el **Análisis Estadístico de Faltas, STAFAN**, [JAI84].

7.2. Procedimientos de generación de test y síntesis del módulo monitor.

En esta sección se presentan dos procedimientos de generación de test para las faltas de anclaje simples. El primero (procedimiento 7.1) se basa en la utilización de la transformada de Tamari, ya que, para seleccionar los patrones de test, se utilizan los coeficientes espectrales de Tamari que cambian cuando se produce una falta dada. Los cambios que una falta puede ocasionar en el espectro de la función que sintetiza el circuito se deducen de los teoremas y corolarios de los capítulos 2 y 3, a partir de los cuales se han elaborado algoritmos que permiten la automatización del proceso. El procedimiento 7.2 se basa en la selección aleatoria de patrones utilizándose (7.20) para determinar el número de patrones que deben constituir el test. A continuación se realiza la descripción de estos procedimientos, donde se utilizará la siguiente notación:

T es el conjunto de patrones de test que se selecciona.

F es el conjunto de faltas de anclaje simple.

FD(C) es el conjunto de faltas detectadas por el test constituido por los patrones contenidos en **C**.

M es el número de faltas.

CF es el cubrimiento de faltas de anclaje simples del test, es decir, $CF = \text{Card}(FD)/M$.

CFmin es el menor valor aceptable para el cubrimiento de faltas.

n_j es el nudo j -ésimo del circuito.

$f_{j,k}$ es la falta de anclaje al valor k ($k=0,1$) en el nudo n_j .

$t(n_j, k)$ es la testeabilidad del valor k ($k=0,1$) en el nudo n_j .

$d(f_{j,k})$ es la detectabilidad de la falta $f_{j,k}$.

L es el número de patrones que constituyen el test, es decir $\text{Card}(T)$.

$R_{j,k}$ es el conjunto de coeficientes espectrales que cambian debido a una falta de anclaje a k ($k=0,1$) en el nudo n_j .

$CP(r_\beta)$ es la complejidad del coeficiente espectral r_β .

$\text{In}(r_\beta)$ es el conjunto de combinaciones de entradas que intervienen en el cálculo del coeficiente espectral r_β . Así, suponiendo que $\beta = (\beta_1, \dots, \beta_m)$, las combinaciones de entradas $Y = (y_1, \dots, y_m)$ que intervienen en la evaluación del coeficiente son las que verifican

$$(y_1 \beta_1 \cup \dots \cup y_{m-1} \beta_{m-1})' = 1.$$

Procedimiento 7.1:

1. $T = \{\emptyset\}$, $FD(T) = \{\emptyset\}$, $CF = 0$.
2. Analizar la testeabilidad del circuito.
3. Seleccionar $f_{j,k} \in (F - FD(T))$ tal que para toda falta $f_{j,k'} \in F - FD(T)$ se verifique que $t(n_j, k') > t(n_j, k)$.
4. Determinar el conjunto $R_{j,k}$.
5. Tomar $r_\beta \in R_{j,k}$ tal que $CP(r_\beta) < CP(r_{\beta'})$ para cualquier $r_{\beta'} \in R_{j,k}$.
6. Obtener $FD(\text{In}(r_\beta))$ y $FD(T) = FD(T) \cup FD(\text{In}(r_\beta))$.
7. $T = T \cup \text{In}(r_\beta)$.
8. Calcular CF. Si $CF > CF_{\text{min}}$, terminar; en caso contrario, ir a 3.

Procedimiento 7.2:

1. $T = \{\emptyset\}$, $FD(T) = \{\emptyset\}$, $CF = 0$.
2. Analizar la testeabilidad del circuito.
3. Seleccionar $f_{j,k} \in (F - FD(T))$ tal que para toda falta $f_{j,k'} \in F - FD(T)$ se verifique $t(n_j, k') > t(n_j, k)$.
4. Determinar la detectabilidad $d(f_{j,k})$ para la falta seleccionada.
5. Utilizando (7.20), calcular el número L de patrones de test necesarios para asegurar un cubrimiento de faltas mayor que CF_{min} .
6. Seleccionar aleatoriamente los L patrones del conjunto T y terminar.

Si se desea asegurar que el conjunto de patrones seleccionados garantiza el porcentaje de cubrimiento mínimo que se ha fijado, el procedimiento 7.2 se puede completar con los siguientes dos pasos, eliminando previamente "terminar" en el paso 6:

Procedimiento 7.2 (continuación):

7. Determinar $FD(T)$ y calcular CF . Si $CF > CF_{min}$ terminar, si no, ir a 8.
8. Aplicar los pasos 2 a 8 del procedimiento 7.1 a las faltas no detectadas. ■

En el paso 2 de los procedimientos 7.1 y 7.2 se habla del análisis de testeabilidad del circuito. En la presente memoria se utiliza el programa SCOAP, pero se puede recurrir a cualquier otro programa de análisis de testeabilidad disponible: CAMELOT, TMEAS, etc (sección 1.2.2).

En el paso 3 de ambos procedimientos también se podrían tomar varias faltas de testeabilidad baja en lugar de una. Eso dependería de la complejidad del circuito y de la capacidad de cálculo de que se disponga. En pasos posteriores, se considerarán los coeficientes espectrales que se modifican por todas esas faltas si se trata del procedimiento 7.1, o se determinarán las detectabilidades de todas las faltas seleccionadas si se utiliza el 7.2.

En el paso 4 del procedimiento 7.1 se selecciona un coeficiente espectral que cambie debido a la falta que se está considerando. Para ello, a) se pueden aplicar los teoremas y corolarios que, para distintas estructuras de circuitos, se dan en la sección 3.2; b) se puede obtener el espectro para la función que sintetiza el circuito afectado por la falta y compararlo con el de la función correcta; o c) aplicar la heurística dada en la sección 3.3.

La comprobación de las faltas detectadas por los patrones de test se ha realizado con el simulador de faltas que proporciona el paquete software LESIM2, para el diseño de ASICs.

La detectabilidad de una falta, necesario en el paso 4 del procedimiento 7.2, se puede calcular mediante simulación de dicha falta para todas las posibles combinaciones de entradas o bien, si el número de entradas es demasiado elevado, mediante técnicas estadísticas como pueden ser el programa STAFAN o el muestreo de patrones de entradas, o métodos algebraicos.

Por consiguiente, el procedimiento 7.2 se puede considerar un proceso previo al 7.1 que posiblemente permita una selección rápida de patrones. Si el resultado de 7.2 no es satisfactorio, se puede continuar con 7.1, o bien olvidarse de los resultados de 7.2 y empezar por 7.1 de nuevo.

En los dos procedimientos de selección se presupone que una cobertura de faltas simples es suficiente para asegurar una relación de rechazo satisfactoria. Como se vió en el capítulo 1, esta suposición está justificada por los resultados presentados por diversos autores [ABR80, BRE76, HUG86, JAC87, JHA86] y se realiza en todos los procedimientos de selección y evaluación de patrones [FUJ83, GOE81, MAL84, ROT67, WAG87]. Por otra parte, hay que tener en cuenta que el procedimiento es válido cualquiera sea el modelo de faltas y el nivel de descripción utilizado. Sólo se necesita disponer de programas para determinar la testeabilidad a ese nivel de descripción y para simular las faltas correspondientes. Además, en el caso del procedimiento 7.1 son necesarios los criterios para determinar qué coeficientes espectrales cambian dada una de esas faltas. En el capítulo 2 se proporcionan estos criterios para el caso de faltas de anclaje y en el apéndice IV para las faltas a nivel de transistor en circuitos CMOS.

Si al utilizar el procedimiento 7.1 se obtuviera un cubrimiento de faltas aceptable con un sólo coeficiente espectral, a partir de dicho coeficiente se podría sintetizar un compresor de salidas. Por consiguiente, el procedimiento 7.1 también resulta útil si se ha pensado en una solución tipo BIST off-line para el test del circuito.

Como ejemplos de aplicación de estos procedimientos, se va a seleccionar un conjunto de patrones para testear todas las faltas de anclaje del circuito 74181, que es una ALU de 4 bits cuyo esquema se muestra en la figura 7.1. Se trata de un circuito de 14 entradas, 8 salidas y 200 nudos, que se utiliza con mucha frecuencia como ejemplo en la literatura de test [CHI87]. La detectabilidad más baja para las faltas de anclaje simple de este circuito es de 96 patrones [CHI87] y además se le ha aplicado el programa SCOAP para analizar su testeabilidad (los resultados se muestran en el Apéndice VI). Utilizando el valor de detectabilidad mínima, a partir de (7.20) se puede estimar la longitud del test aleatorio necesario. Así, si se desea que la peor falta se detecte con el 0.9 de probabilidad, $c_L(i)=0.9$, la longitud del test será $L=\log(1-0.9)/\log(1-(96/16384))=392$ patrones. En la figura 7.2 se muestra la variación del porcentaje de faltas detectadas a medida que se van aplicando los patrones de un test de longitud $L=417$ ($L>392$) seleccionado mediante el procedimiento 7.2. El porcentaje de faltas detectadas al final del test es del 97.6%. Como se puede observar, en el caso de la selección aleatoria de patrones se consigue un cubrimiento elevado de faltas con pocos patrones, pero llega un momento en el que el incremento de cubrimiento de faltas prácticamente se estanca. Esta situación se produce debido a la presencia de

faltas con baja detectabilidad que hacen poco probable su detección si se han seleccionado pocos patrones.

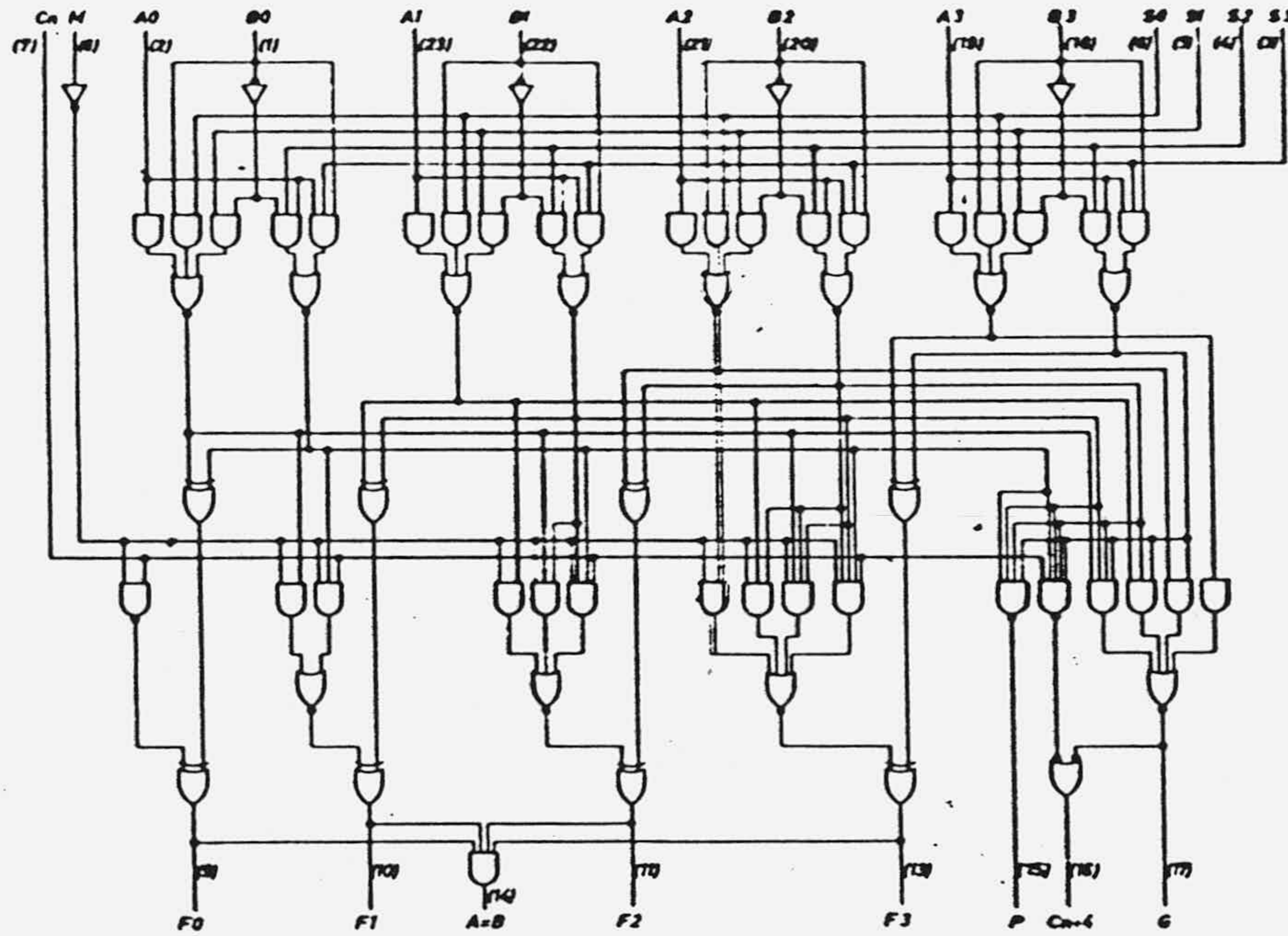


Figura 7.1: Esquema de la ALU 74181.

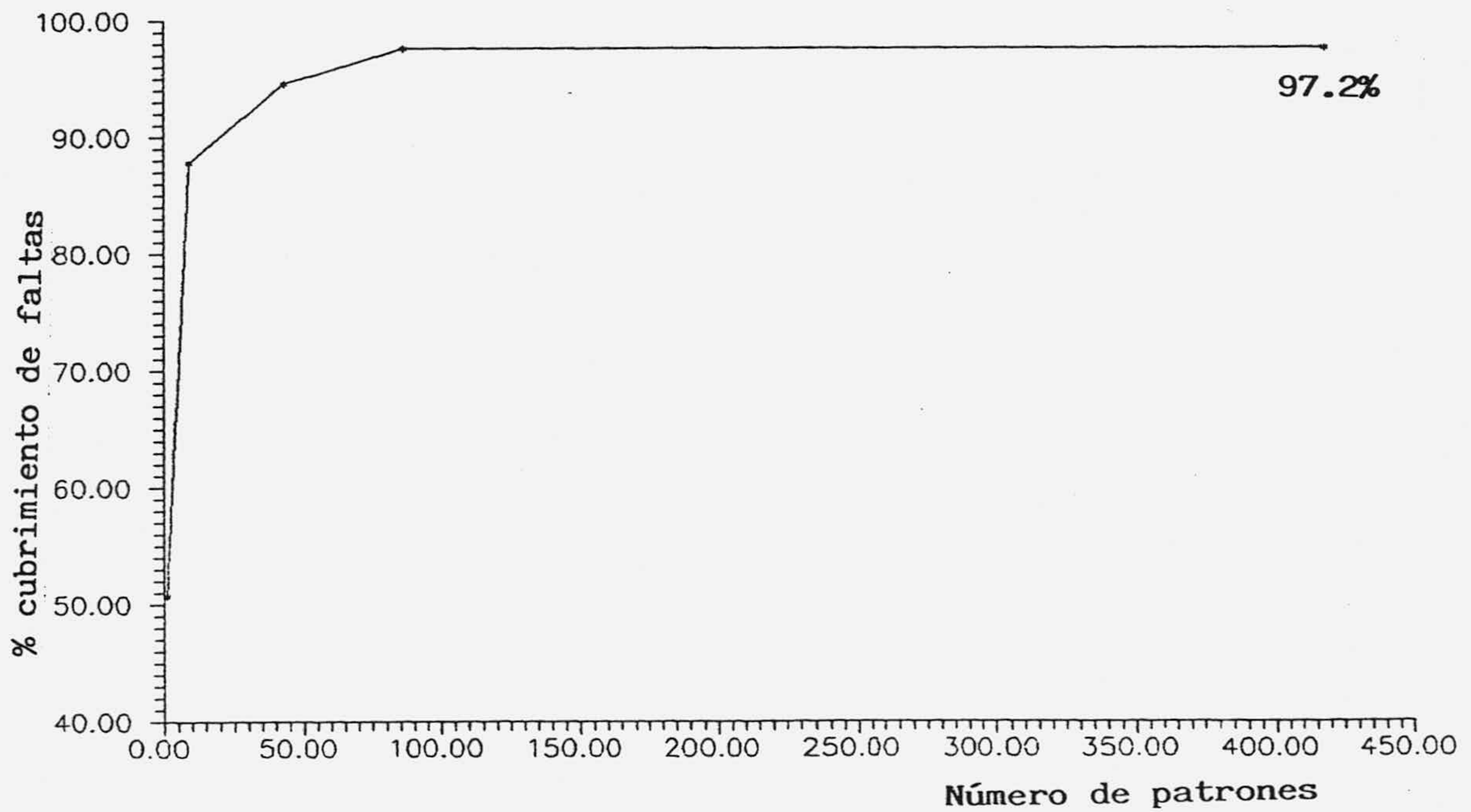


Figura 7.2: Porcentaje de faltas detectadas.

Las figuras 7.3 y 7.4 corresponden al proceso de selección de patrones mediante el procedimiento 7.1. Muestran el incremento del cubrimiento de faltas frente al número de coeficientes espectrales seleccionados para determinar el test (figura 7.3) y frente al número de patrones que se van añadiendo al test al ir seleccionando nuevos coeficientes (figura 7.4). En la tabla 7.1 se indica el orden de selección de los coeficientes espectrales y el índice de testabilidad del nudo que se ha utilizado para seleccionar el coeficiente espectral. El índice de testabilidad se obtiene a partir de los resultados de SCOAP tal y como se define en [AGR88], es decir a partir de la suma $t=2co+cc0+cc1$, donde co es la observabilidad combinacional, $cc0$ y $cc1$ son las controlabilidades combinacionales del 0 y del 1, respectivamente. Como se puede observar en la figura 7.3, con 35 coeficientes espectrales se define el conjunto de test que está formado por 417 patrones, con los que se consigue detectar el 100% de las faltas de anclaje. Interesa tener en cuenta que, para almacenar el test a través de los coeficientes espectrales, se precisarían $35 \times 14 = 490$ bits de memoria, frente a los $417 \times 14 = 5838$ que se necesitarían para almacenarlo patrón a patrón. En la figura 7.5 se comparan los resultados del procedimiento 7.1 y los del procedimiento 7.2. Como se puede comprobar, aunque se consiga antes el 100% de cubrimiento de faltas para el test seleccionado por el procedimiento 7.1, el cubrimiento de faltas que consiguen los patrones seleccionados por el procedimiento 7.2 es mayor para menos de 399 patrones aproximadamente. En ese sentido, puede ser interesante aplicar la continuación del procedimiento 7.2.

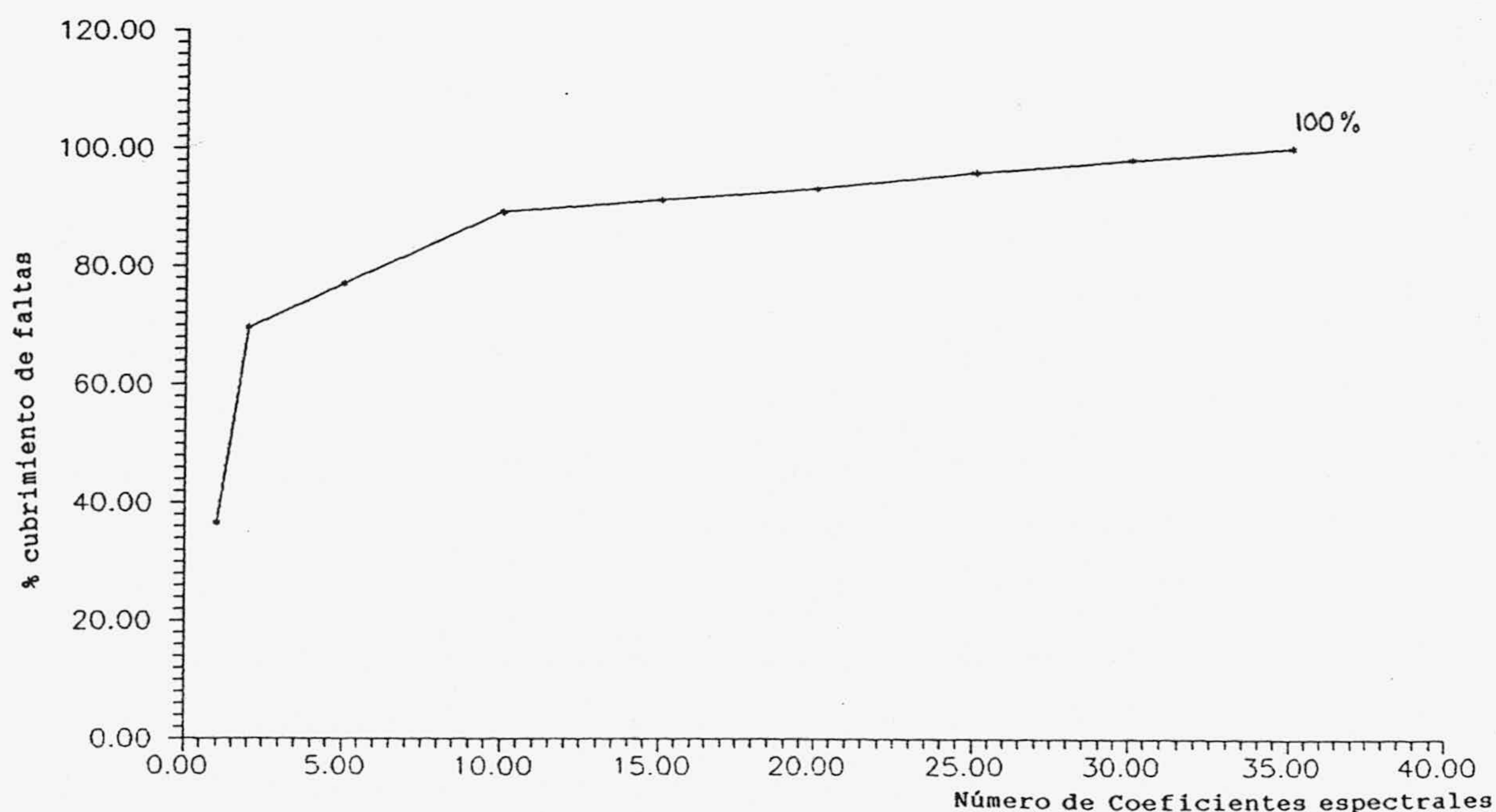


Figura 7.3: Cubrimiento de faltas frente a coeficientes espectrales.

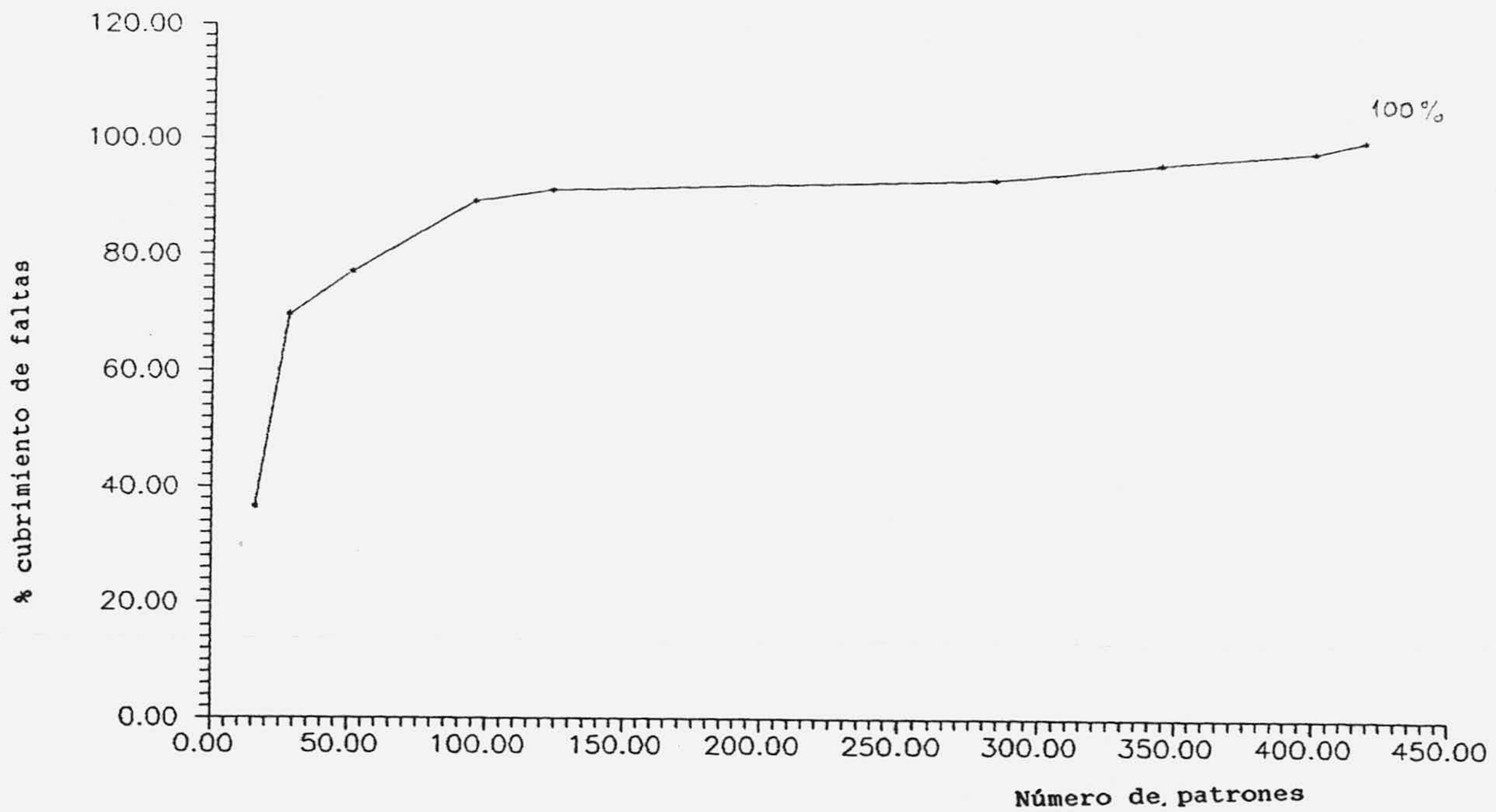


Figura 7.4: Cubrimiento de faltas frente a patrones.

Tabla 7.1: Orden de selección de los coeficientes espectrales.

Nudo	Coficiente espectral	Indice de testabilidad
N48	C(6192)	76
N20	C(12832)	76
N1	C(2880)	76
N2	C(13056)	70
N55	C(12336)	60
N67	C(6209)	59
N63	C(6401)	54
N35	C(8242)	53
N62	C(8386)	48
N60	C(2069)	48
N36	C(10264)	47
N5	C(12808)	46
N4	C(12336)	44
N9	C(2316)	44
N14	C(10816)	43
N7	C(12300)	42
N11	C(12802)	42
N31	C(365)	42
N29	C(12724)	42
N44	C(6912)	41
N0	C(12480)	40
N45	C(6336)	39
N41	C(857)	39
N25	C(12801)	38
N28	C(5312)	38
N37	C(2061)	37
N50	C(5168)	36
N65	C(8206)	34
N30	C(9804)	32
N42	C(8288)	32
N15	C(5888)	30
N23	C(12324)	26
N32	C(1038)	24
N66	C(8203)	24
N18	C(12928)	18

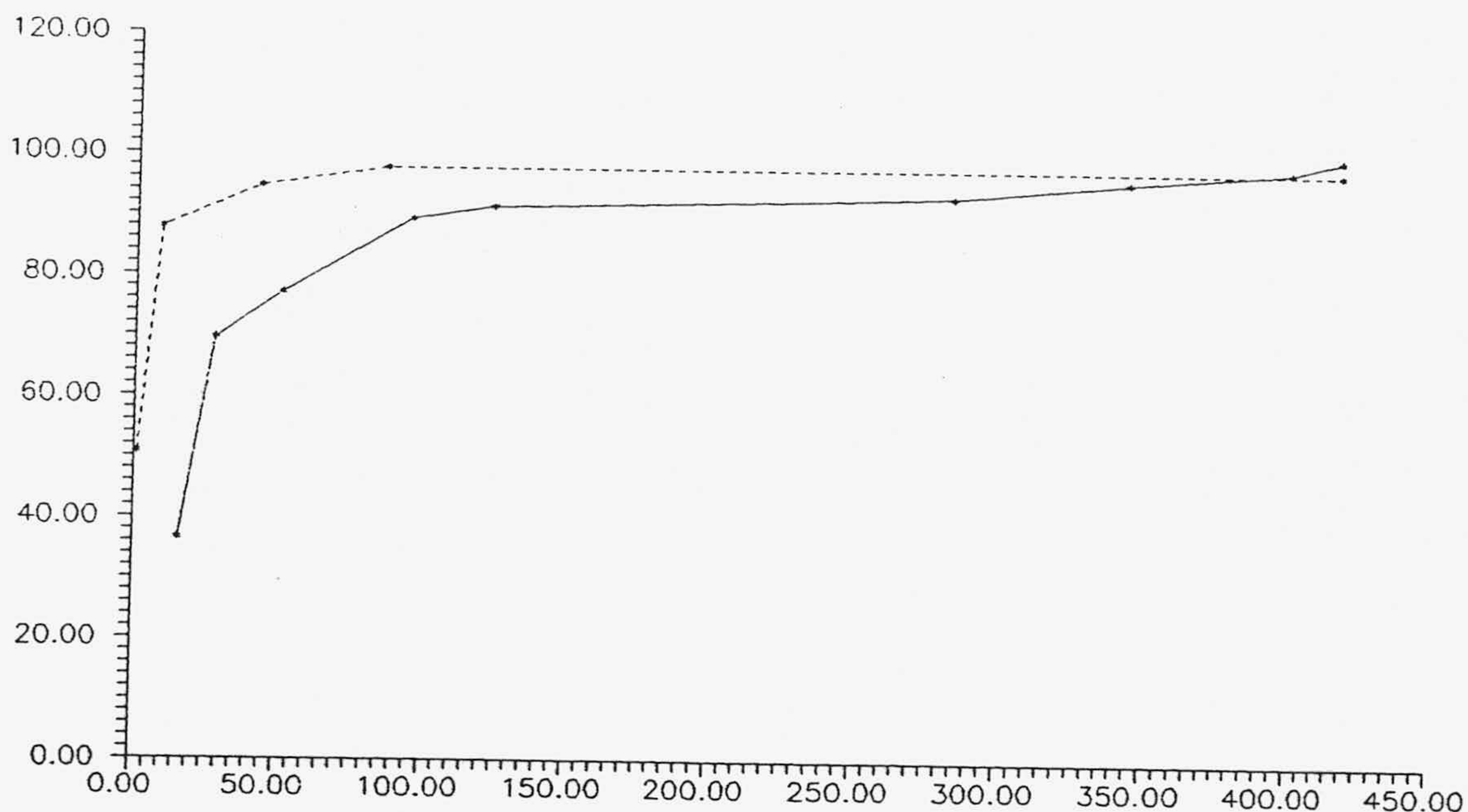


Figura 7.5: Comparación de los procedimientos 7.1 y 7.2.

La figura 7.6 muestra el cubrimiento de faltas para la ALU 74181 frente al número de patrones seleccionados aplicados, cuando se utiliza un procedimiento de selección aleatoria para alcanzar con pocos patrones un cubrimiento de faltas aceptable y añadir los patrones que permiten la detección de las faltas restantes mediante el procedimiento determinista 7.1.

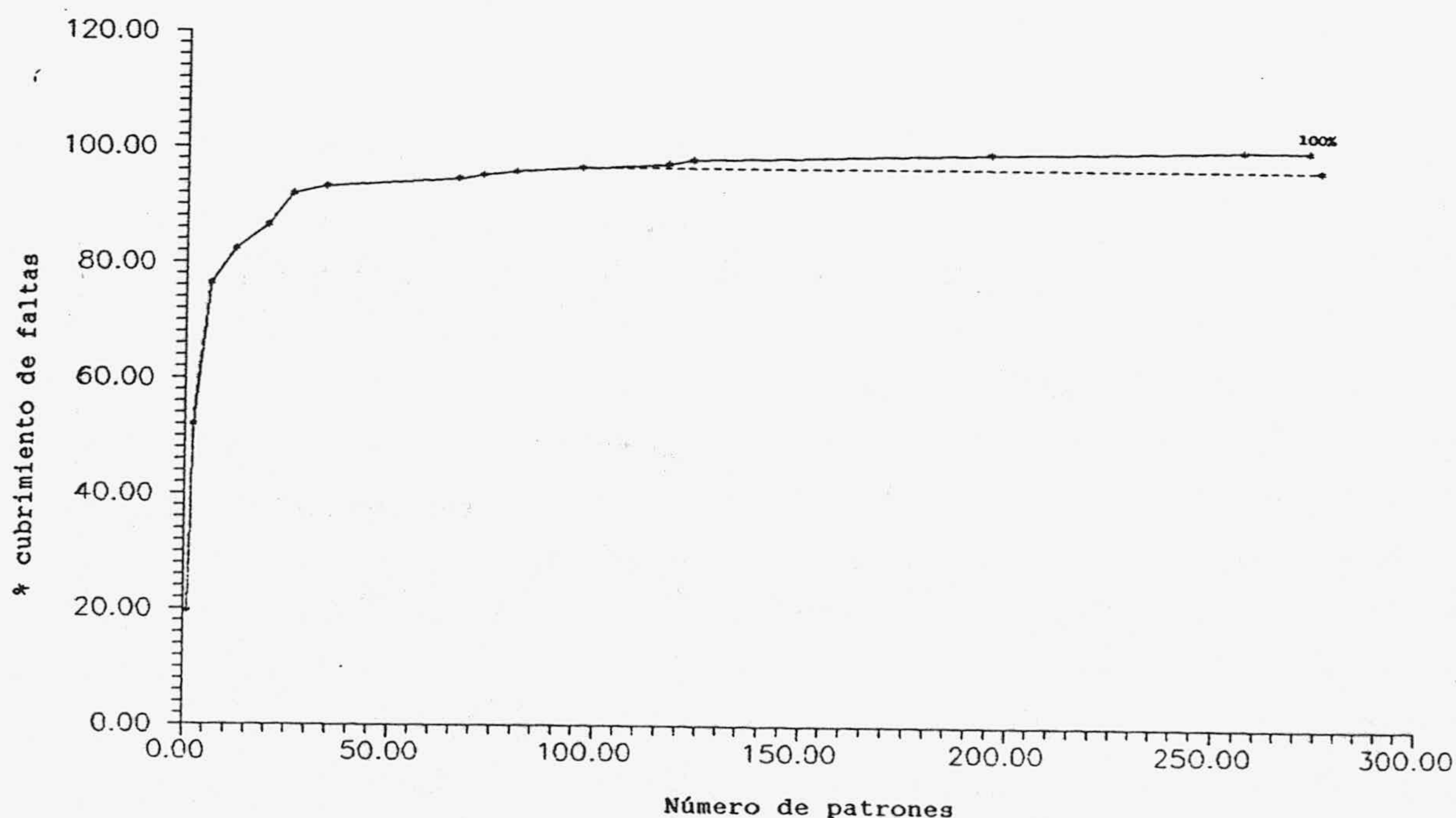


Figura 7.6: Cubrimiento de faltas para la ALU 74181.

En concreto, con el procedimiento de selección aleatoria se consigue un 96.6% de cobertura de faltas. Quedan 5 faltas sin cubrir para las que se determinan 5 coeficientes espectrales que permiten la selección de los patrones adicionales que las detectan: $c(13056)$, $c(365)$, $c(857)$, $c(8288)$, $c(9804)$. La línea a trazos corresponde a la selección aleatoria. Obsérvese que se consigue el 100% de cubrimiento con 274 patrones frente a los 417 necesarios si se utiliza el procedimiento determinista únicamente (ver la figura 7.4).

7.2.1. Algoritmo de selección aleatoria de patrones.

El problema que se puede plantear cuando se eligen aleatoriamente los patrones que constituyen el test T (paso 6 del procedimiento 7.2) es que la síntesis de la función Θ_T puede dar lugar a un monitor excesivamente complejo.

En esta sección se presenta un algoritmo para la selección aleatoria de los L patrones, $L=L(m)2^{m-1}+\dots+L(2)2+L(1)$, que han de constituir el test de manera que la complejidad del módulo monitor de patrones sea aceptable. La clave del algoritmo estriba en seleccionar aleatoriamente los términos de la expresión de Θ_T como suma EXOR de las variables de entrada complementadas o sin complementar, de tal forma que para cada combinación de entradas, a lo sumo uno de los términos de la sumatoria se haga igual a uno. Es decir, la expresión de $\Theta_T(y_1, y_2, \dots, y_m)$ tendrá la forma

$$\Theta_T(Y) = C(m) \oplus C(m-1) \oplus \dots \oplus C(2) \oplus C(1)$$

donde $C(i)=0$ si $L(i)=0$, o un producto con $m-i+1$ variables complementadas o sin complementar construido según el algoritmo 7.1. En dicho algoritmo, el término $C(i)$ se construye complementando alguna de las variables del término $C(j)$, $j < i$, distinto de cero que tenga el índice j más próximo al índice i y añadiendo algunas variables nuevas seleccionadas aleatoriamente. El algoritmo es como sigue:

Algoritmo 7.1:

1. $j=0$, $w=0$.
2. Hacer $i=m-j$. Si $i=0$ terminar.
3. Si $L(i)=0$, hacer $j=j+1$, $C(i)=0$ e ir a 2.
4. Si $w \neq 0$ ir a 8.

5. Hacer $C(i) = y_{k_1}^{j_1} \dots y_{k_r}^{j_r}$, donde $r = m - i + 1$, $j_s \in \{0, 1\}$, $k_s \in \{1, \dots, m\}$, $y^0 = y$, $y^1 = y'$ y los valores de k_s y j_s se eligen aleatoriamente. El número de unos en Θ_T a que da lugar este término es, precisamente, 2^{m-i-1} .
6. Definir el producto $V(w) = C(i)$.
7. $w = w + 1$, $j = j + 1$. Ir a 2.
8. Hacer $C(i) = V(w-1) * y_{k_1}^{j_1} \dots y_{k_s}^{j_s}$, donde $V(w-1)^*$ coincide con $V(w-1)$ salvo en algunas variables, elegidas aleatoriamente, que han cambiado su situación respecto a estar complementadas o no; los índices $k_r \in \{0, \dots, m-1\}$ no coinciden con ninguno de los índices de $V(w-1)$; y $j_r \in \{0, 1\}$. Los índices k_r y j_r se eligen aleatoriamente. Con este nuevo término se añaden otros 2^{m-r-1} unos.
9. Ir a 6.

Por el proceso de construcción de los términos, al final se tienen exactamente L unos en la función Θ_T . Además, aunque la forma de los términos está determinada previamente, la construcción de cada uno de ellos se hace seleccionando aleatoriamente las variables que intervienen, por lo que se puede hablar de una cierta selección aleatoria de patrones. Por otra parte, el número de términos que aparece en la suma EXOR es menor o igual que m : depende del número de unos que haya en la expresión binaria de L .

Como ejemplo, sea un circuito con 6 entradas $y_0, y_1, y_2, y_3, y_4, y_5$, para el que se quieren seleccionar 13 patrones aleatoriamente. En este caso $L = 001101$ por lo que $\Theta_T = C(4) \oplus C(3) \oplus C(1)$, y una posible asignación para esos términos es

$$C(4) = y_1 y_2 y_3', \quad C(3) = y_1 y_2' y_3' y_4 \quad \text{y} \quad C(1) = y_1 y_2' y_3' y_4' y_0' y_5.$$

A continuación se comprueban los resultados de simulación obtenidos con L patrones seleccionados según este algoritmo y con L patrones seleccionados aleatoriamente tomando como ejemplos el circuito de la figura 3.9 y la ALU 74181. En la figura 7.7 se muestra el cubrimiento de faltas frente al número de patrones para la ALU74181. La línea continua corresponde a la selección de patrones con el algoritmo 7.1 y la discontinua a la selección aleatoria. El comportamiento de ambas curvas es similar, existiendo un incremento más rápido del cubrimiento al principio y un estancamiento a medida que las faltas de mayor detectabilidad se cubren. Sin embargo el crecimiento del cubrimiento para la secuencia seleccionada con el algoritmo 7.1 es menor y el estancamiento se produce para un valor de cubrimiento más bajo.

No obstante, el algoritmo 7.1 se puede utilizar si permitiera alcanzar un cubrimiento de faltas elevado con pocos patrones añadiendo el resto mediante el procedimiento 7.1 de selección determinista.



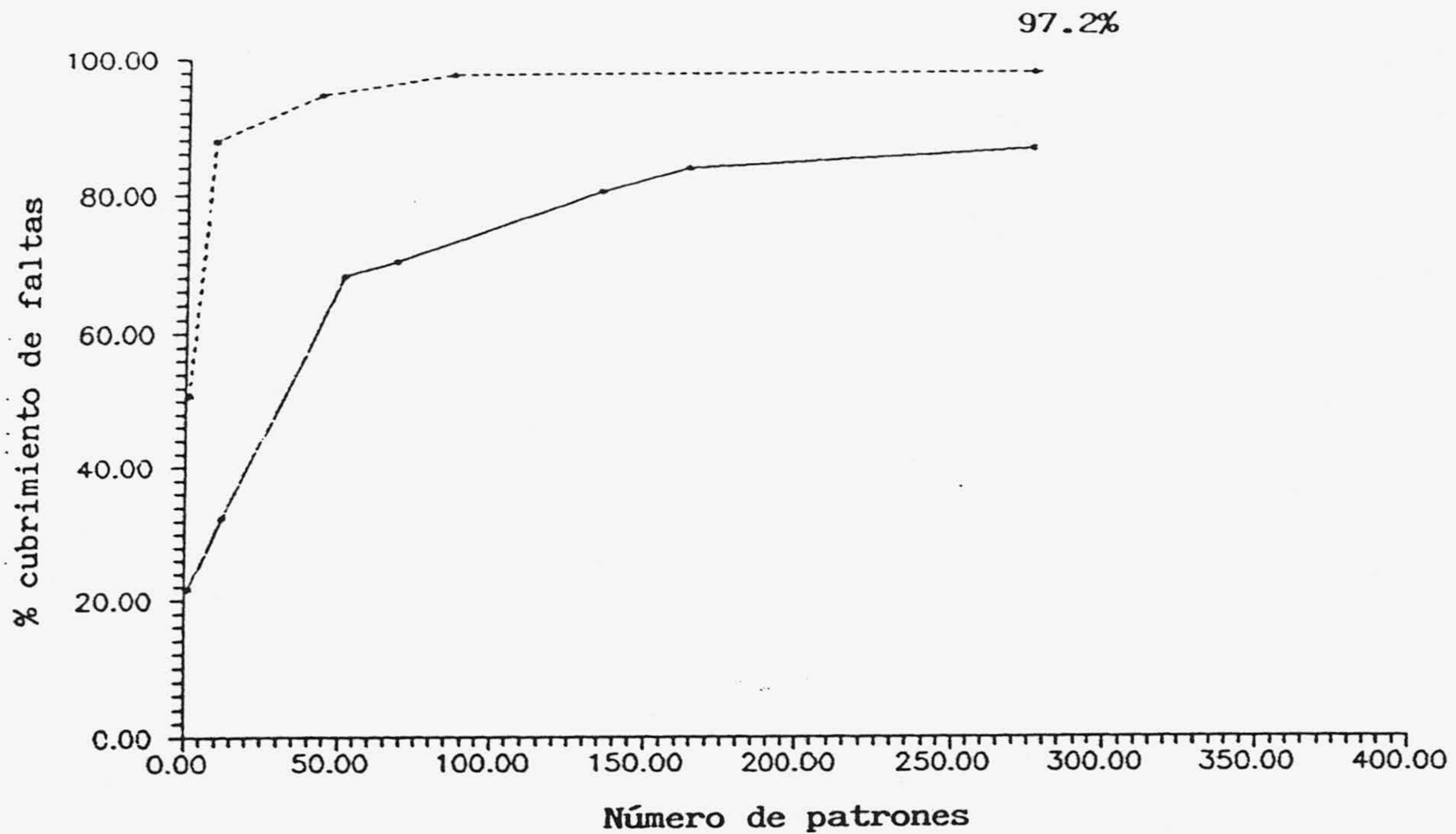


Figura 7.7: Cubrimiento de faltas (ALU 74181).

7.3. Síntesis del circuito predictor.

La función que ha de sintetizar el predictor queda determinada una vez fijados los coeficientes de Tamari que permiten la síntesis del extractor, que sintetiza r funciones $G=(g_1, \dots, g_r)$, $g_i \in F_{2^N}$ ($i=1, \dots, r$), cuyas N entradas $X=(x_1, \dots, x_N)$ corresponden a las salidas del circuito a testear, R^F . La determinación de las funciones del extractor se puede realizar mediante el algoritmo basado en el recocido simulado, expuesto en el capítulo 6. Por tanto, el circuito R^F y el extractor sintetizan r funciones, $h_i \in F_{2^m}$, de m variables $(y_0, y_1, \dots, y_{m-1})$, notadas como $H=(h_1, \dots, h_r)$:

$$H(Y)=G(F(Y)) \quad (7.25)$$

Las salidas del extractor y del predictor constituyen las entradas del comparador, que para toda entrada Y al circuito da una salida E que indica si se ha producido una respuesta correcta o errónea para la entrada aplicada. Teniendo en cuenta que el comparador debe ser totalmente autocomprobante, la función que debe realizar el módulo predictor es la complementaria de H , $PR(Y)=H'(Y)$, es decir

$$\begin{aligned} E &= \text{No Error si } \forall i=1, 2, \dots, r \quad g_i(Y)=1 \oplus h_i(Y) \\ & \text{Error en otro caso.} \end{aligned} \quad (7.26)$$

donde los valores concretos que se asocien a la salida E para indicar que ha habido error o no lo ha habido dependeran de la forma de realizar el comparador. En ese sentido se pueden consultar algunos de los trabajos publicados con respecto a la síntesis de circuitos comparadores totalmente autocomprobantes, especialmente [BOS84d, HUG84, JHA89, PIE87, TAM84]. Por ejemplo, si se utiliza un comparador de dos vías (Two-Rail), la situación de Error se indica con E=00 ó E=11, y la situación de No Error con E=01 ó E=10.

Por otra parte, al diseñar la circuitería para el test concurrente se debe buscar que su complejidad sea la menor posible. En la síntesis del módulo extractor esto se tiene en cuenta al buscarse el número mínimo de funciones de extracción que aseguran una probabilidad de enmascaramiento mínima. La minimización de la circuitería del predictor se hace a partir de la selección de patrones de test que lleva a cabo el monitor. Este indica qué entradas son patrones de test y sólo cuando se aplica una de éstas se valida el resultado del comparador. Sólo para las entradas que correspondan a patrones de test está fijada la salida del módulo predictor; el resto de salidas son indiferencias, notadas con d, que permiten reducir la complejidad del módulo predictor. Así, este módulo predictor realizará la función:

$$\begin{aligned} PR(Y) &= H'(Y) \text{ si } Y \in T \\ & d \text{ en caso contrario.} \end{aligned} \quad (7.27)$$

Para obtener la expresión mínima del módulo predictor, se precisan algoritmos de síntesis multifuncional mínima. Si la síntesis se pretende realizar mediante una PLA del tipo AND-OR, se puede utilizar el ESPRESSO; en cambio, si la síntesis se va a realizar con una PLA AND-EXOR, puede servir el procedimiento presentado en esta memoria, descrito en el Apéndice II. No obstante, hay que tener en cuenta que para aprovechar las indiferencias de la función del predictor de cara a reducir la complejidad relativa de la circuitería del test concurrente, hay que utilizar el mismo procedimiento de síntesis que el del circuito a testear. Es decir, si el CUT se realiza mediante una PLA, el incremento de circuitería que se produce es menor que el que correspondería a duplicar la PLA pero no está tan claro que se tenga esa situación si el circuito se ha realizado con lógica aleatoria, a no ser que se ataque la síntesis del módulo predictor de la misma forma.

Aunque, como se ha dicho, el objetivo primordial al llevar a cabo una selección de entradas como patrones de test es la simplificación del módulo predictor, también contribuye a solucionar otro problema: si la función que implementa el circuito no se conoce, se puede determinar mediante simulación

de entradas es elevado, el tiempo que se invertiría en la simulación y, sobre todo la cantidad de memoria que se necesitaría, serían demasiado elevados. Si la simulación se realiza sólo para el conjunto de entradas seleccionadas como patrones de test, se solucionan las dificultades.

7.3.1. Ejemplo de diseño para el test concurrente.

En esta sección se completa el diseño para el test concurrente del circuito de la figura 6.3 para el que se obtuvieron las funciones de extracción en la sección 6.4. En primer lugar, el conjunto de patrones de test para las funciones $x_1=f_1(y_1, \dots, y_5)$, $x_2=f_2(y_1, \dots, y_5)$ y $x_3=f_3(y_1, \dots, y_5)$ que implementa se seleccionan a partir de los coeficientes espectrales, $c(j_1, \dots, j_5)$, $c(11000)$, $c(01100)$, $c(00101)$ y $c(00011)$. Estos coeficientes espectrales definen un total de 10 patrones de test para los que se consigue un cubrimiento de faltas del 100%. En la figura 7.8 se muestra el circuito monitor de patrones que definen esos coeficientes y que permite validar la salida del módulo comparador. El extractor sintetiza la función obtenida en la sección 6.4, $g_1(x_1, x_2, x_3)=x_3 \oplus x_2 \oplus x_1 x_3$, para la que la probabilidad de enmascaramiento asociada es igual a $p_A=0.133$.

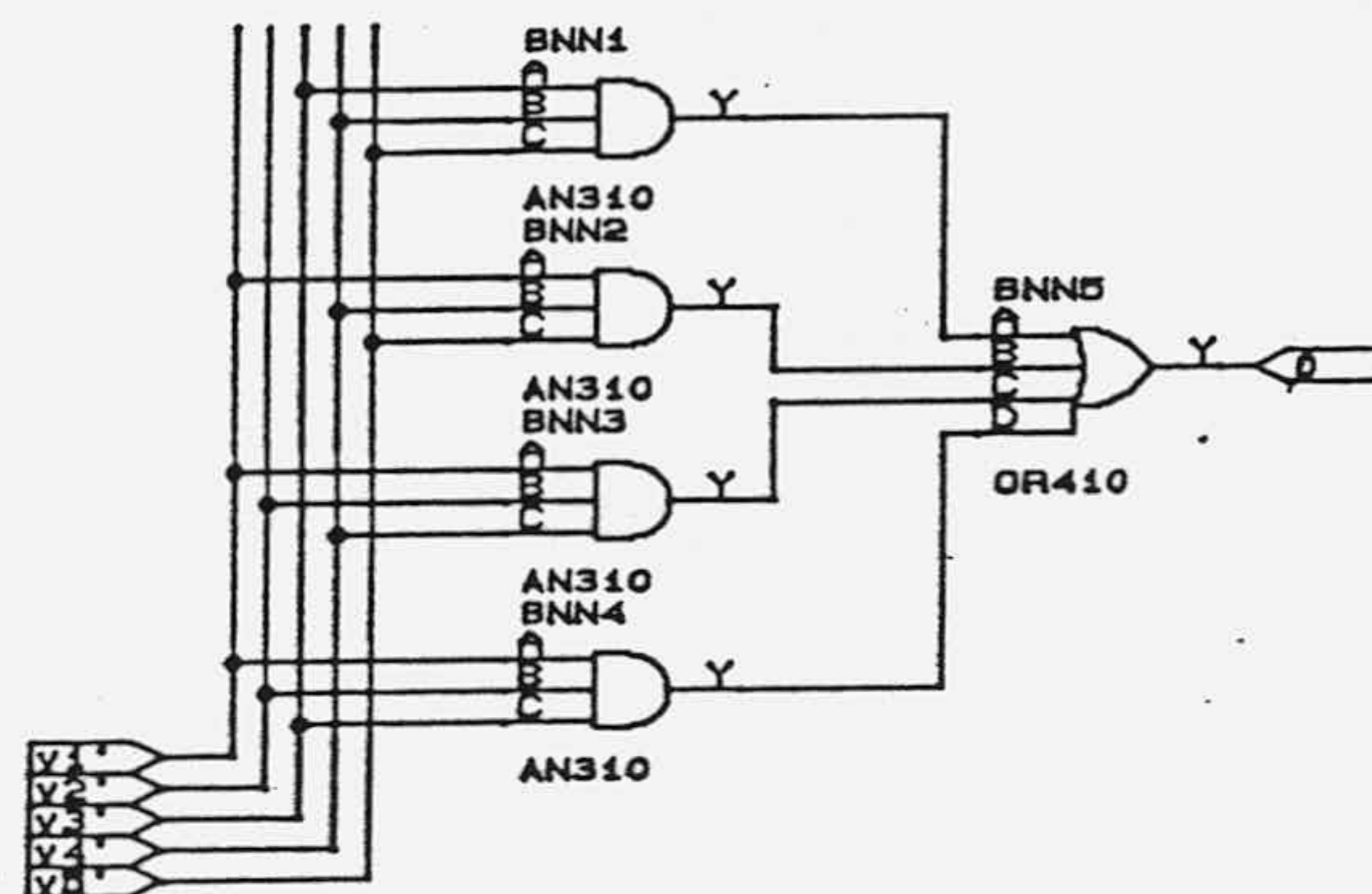


Figura 7.8: Circuito monitor de patrones del ejemplo.

Según (7.27), la función que ha de realizar el bloque de predicción ha de coincidir con la complementaria de la que se obtiene a la salida del módulo extractor para los patrones de test. En la figura 7.9 se muestra el esquema del circuito con el módulo de extracción y el módulo de predicción.

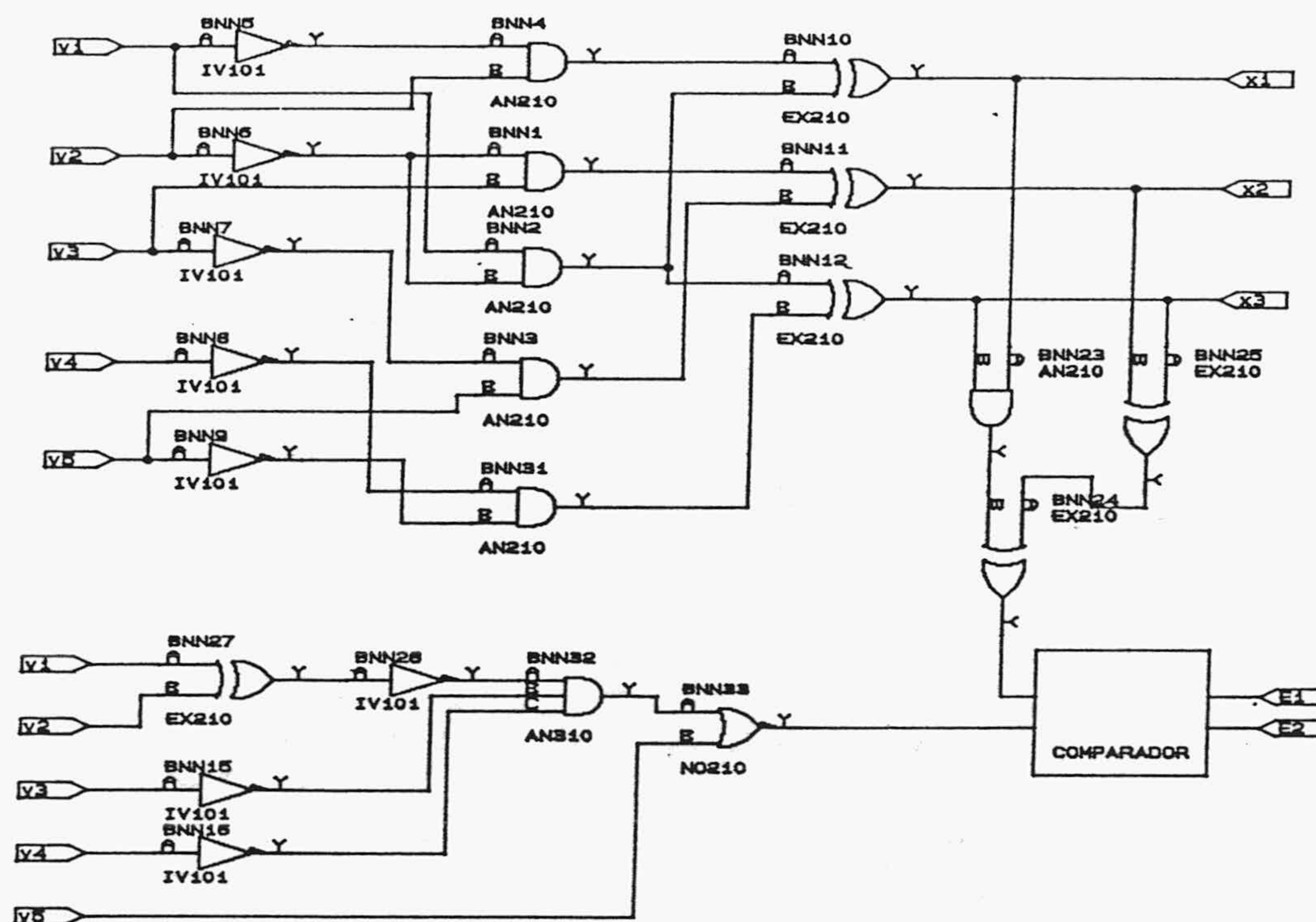


Figura 7.9: Esquema de los módulos de extracción y de predicción.

Cuando se aplican los patrones de test al circuito y se comprueba únicamente el estado del bit de error puede ocurrir que no todos las faltas se sigan detectando debido a que el extractor enmascare el efecto de todos los patrones para una falta determinada. En este caso, tal y como se indica en la sección 7.1, habría que aumentar el número de patrones de test para conseguir un 100% de cubrimiento de faltas para el circuito con el módulo extractor, y considerando como salidas observables únicamente las de dicho módulo.

En el ejemplo que se está considerando aquí, no hay que añadir patrones nuevos, puesto que con los ya seleccionados se consigue un 100% de cubrimiento de las faltas simples de anclaje.

Para evaluar el tiempo de latencia de un defecto que se modele como falta de anclaje simple, se ha determinado en primer lugar la probabilidad de que se produzca un error en las salidas del circuito utilizando la expresión (7.1), que depende de la probabilidad de aplicar una entrada $p(Y)$ y de la probabilidad de que cuando se aplique un patrón, Y , se produzca un error E .

Las probabilidades $p(Y)$ se considerarán iguales para todo patrón de test e iguales para todas las entradas posibles, por lo que $p(Y)=1/32$. Entonces (7.1) se puede escribir como

$$p_e = 1/32 (\sum_{Y \in T} (\sum_{E \neq 0} p(E/Y)))$$

y la suma $\sum_{E \neq 0} p(E/Y)$ para cada patrón Y corresponde a la fracción de faltas que dicho patrón puede poner de manifiesto. Por consiguiente se pueden obtener por simulación, estableciendo el cubrimiento de faltas de un test formado por un sólo patrón, el patrón Y correspondiente. En el ejemplo que se está considerando, el valor obtenido $p_e=0.124$. Utilizando los valores de p_A y p_e el valor del tiempo de latencia previsto es $T_L=1/p_e(1-p_A)=9$. En la Tabla 7.2 se indican los valores de las sumatorias $\sum_{E \neq 0} p(E/Y)$ obtenidas para cada patrón de test Y .

Tabla 7.2: Fracción de faltas de anclaje detectadas por cada patrón.

Y	$\sum p(E/Y) (E \neq 0)$
00000	0.455
10000	0.418
01000	0.418
11000	0.491
00100	0.418
01100	0.400
00001	0.327
00101	0.345
00010	0.382
00011	0.309

El tiempo de latencia también se puede evaluar de forma experimental, mediante simulación del comportamiento del circuito cuando se ve afectado por una falta y aplicando entradas seleccionadas al azar. En la Tabla 7.3 se dan los resultados obtenidos para la simulación de tres secuencias de combinaciones de entradas aleatorias y en cuatro casos de circuito afectado por una falta de anclaje simple. En la tabla se indican la proporción de patrones de test frente al número de entradas y el lugar que ocupan en la secuencia aplicada. En cada uno de los casos se indica el número de combinación de entrada, $N(e)$, que origina la primera combinación de salidas errónea y el número de combinación de entrada, $N(d)$, (que ha de corresponder

a un patrón de test) para la que se detecta el error (en E). El valor de $N(d)$ es proporcional al tiempo de latencia, puesto que se supone que los patrones empiezan a contabilizarse cuando se produce el defecto en el circuito, y el valor de $N(d)-N(e)$ es proporcional al tiempo que pasa desde que se produce la primera salida errónea debido al defecto hasta que se detecta el defecto. A partir de los resultados obtenidos, el valor medio de $N(e)$ es 6.43 ± 2.44 . Por consiguiente se ha obtenido para esta muestra de faltas y secuencias un tiempo de latencia del orden del calculado a partir de la expresión (7.1).

Tabla 7.3: Simulación de tres secuencias aleatorias de combinaciones de entradas.

FALTA	SECUENCIA ALEATORIA	$N(e)$	$N(d)$	$N(d)-N(e)$
BNN1	A.1	4	7	3
	A.2	6	6	0
	A.3	12	12	0
BNNS1/1	A.1	2	2	0
	A.2	2	3	1
	A.3	2	2	0
BNN4/0	A.1	11	12	1
	A.2	1	1	0
	A.3	3	11	8
BNN2/1	A.1	2	3	1
	A.2	1	6	5
	A.3	1	12	11

SECUENCIA ALEATORIA	POSICION DE PATRONES DE TEST EN SECUENCIA	Patrones test/ /Entrada secia.
A.1	1, 2, 3, 5, 7, 10, 12, 13, 20, 21	10/32
A.2	1, 3, 6, 10, 14, 23, 25, 27, 28	9/32
A.3	2, 8, 11, 12, 13, 16, 18, 20, 27, 28, 30	11/32

7.4. Conclusión.

En este capítulo se ha descrito la síntesis del módulo monitor de patrones y del módulo predictor. La base del proceso de síntesis del módulo monitor se encuentra en la selección de patrones. Se han presentado dos métodos de selección de los que uno se puede incluir dentro de los métodos deterministas y está basado en la utilización de los coeficientes espectrales de Tamari, el otro es un método pseudoaleatorio. También se ha propuesto una forma de selección aleatoria de los patrones que permite una síntesis simple del módulo monitor.

Una vez se han determinado los módulo extractor y monitor de patrones, quedan fijadas las funciones que ha de sintetizar el módulo predictor. Para obtener una síntesis mínima de éstas se puede utilizar el ESPRESSO o bien el método propuesto en el Apéndice II.

CAPITULO 8. PRINCIPALES APORTACIONES Y CONSIDERACIONES FINALES.

El trabajo presentado en esta memoria ha tenido como objetivo el desarrollo de una transformación espectral y su aplicación al test de circuitos digitales en sus aspectos de caracterización de defectos como cambios en los coeficientes espectrales, selección de patrones de test, y test concurrente de circuitos. Del estudio realizado sobre los problemas del test de los circuitos integrados digitales se deduce que, dado que para muchos de los problemas planteados no existen algoritmos con complejidad polinómica que permitan encontrar la solución óptima, no es posible plantear un procedimiento de test como el mejor para todos los casos. Habrá técnicas aplicables con mayor eficiencia que otras para cada circuito. Aquí se ha pretendido desarrollar una herramienta para facilitar el test de los circuitos lógicos basada en el uso de una transformación espectral desarrollada por nosotros y a la que se ha dado el nombre de Transformada de Tamari puesto que fue este autor quién estableció que las funciones que se han utilizado para definir la transformación constituyen una base del espacio vectorial $V(F_p^m, C_p, \#, *)$. Esta transformación es aplicable a funciones multivaluadas, siempre que el número de valores posibles sea primo. Una característica importante de esta transformación es que los coeficientes espectrales toman valores dentro del mismo conjunto que la función que se transforma y, como consecuencia, se pueden utilizar las mismas técnicas para la síntesis de circuitos que trabajen en el dominio de la función o en el de su transformada y son análogas las necesidades de memoria para almacenar una función o su espectro de Tamari.

Esto no ocurre con otras transformaciones utilizadas en el contexto de las funciones digitales como la de Walsh, Haar, Fourier, etc.

La contribución del estudio exhaustivo de esta transformación y de su aplicación al test de circuitos digitales, se concreta en las siguientes aportaciones :

a) Se ha desarrollado un procedimiento rápido para obtener la transformación, al que se ha llamado Transformada Rápida de Tamari, ó FTT.

b) Se ha comprobado que cada coeficiente espectral depende de determinados valores de la función. En ese sentido se han establecido la dependencia entre coeficiente espectral y valores de la función de los que depende.

c) Se ha elaborado un algoritmo que permite el cálculo "on-line" de los coeficientes de la transformada de Tamari en el caso binario, es decir se van obteniendo los coeficientes espectrales a medida que se van recibiendo los valores de la función. Ese algoritmo permite obtener un coeficiente espectral cada vez que llega un valor de la función.

d) Se han establecido los cambios que pueden sufrir los coeficientes espectrales cuando los circuitos que realizan las correspondientes funciones sufren ciertos defectos. Se han estudiado aquellos casos en los que los defectos se pueden modelar como faltas de anclaje simples y cortocircuito en las entradas para el caso de circuitos descritos mediante puertas lógicas, como faltas de contacto en PLAs.

e) Se ha elaborado un programa que permite obtener un conjunto de patrones para el test de las faltas simples de anclaje. Dicho programa se basa en una heurística de selección de los coeficientes espectrales, deducida del tipo cambios en los coeficientes espectrales que inducen las faltas, y considerando que si un coeficiente espectral cambia, existe un número impar de patrones de test entre las entradas de la función de las que depende. El programa de selección de patrones que se ha elaborado sólo necesita información acerca de los coeficientes espectrales que intervienen en la función que sintetiza cada nudo del circuito. Esa información se puede obtener mediante una sólo simulación exhaustiva del circuito, sin que sea necesario repetirla para cada falta.

f) Se ha utilizado el procedimiento de selección de patrones a partir de los coeficientes espectrales en combinación con la selección aleatoria de patrones, para reducir el número de patrones que se necesitarían de cara a obtener un cubrimiento completo de las faltas de anclaje si únicamente se utilizara la selección aleatoria de patrones.

g) Se ha elaborado un procedimiento modular para el cálculo de los coeficientes espectrales de un circuito a partir del espectro de los circuitos que lo constituyen y un procedimiento para determinar los coeficientes espectrales que cambian debido a las faltas de anclaje simples a partir del conocimiento de los coeficientes que cambian en cada módulo constituyente del circuito. Un procedimiento de estas características resulta imprescindible para aplicar las técnicas espectrales a circuitos complejos y con un número grande de entradas.

h) Se ha demostrado que el conocimiento de todos los coeficientes espectrales que cambian debido a una falta permite plantear una ecuación booleana a partir de cuya resolución se pueden identificar todos los patrones de test para esa falta.

i) Se ha hecho uso de la transformación espectral para el test concurrente de circuitos. Así, utilizando los coeficientes espectrales de Tamari se han determinado expresiones que permiten evaluar la probabilidad de enmascaramiento de errores debida al módulo extractor y para evaluar la probabilidad de que un conjunto de patrones de test originen un error en las salidas del circuito. A partir de esas medidas se puede evaluar el tiempo de latencia en un circuito que se testea concurrentemente.

j) Se ha propuesto un procedimiento para determinar modelos de error realistas a fin de evaluar de manera fiable la probabilidad de enmascaramiento de un procedimiento de compresión de salidas. En particular, se posibilita la evaluación de la probabilidad de enmascaramiento del módulo extractor.

k) Se ha utilizado un algoritmo, basado en el de Recocido Simulado (Simulated Annealing), para determinar el conjunto de coeficientes espectrales que definen la función que ha de realizar el módulo extractor para obtener un valor de la probabilidad de enmascaramiento fijado, y dado un modelo de error.

l) Se ha propuesto utilizar la selección de patrones de test de cara a introducir indiferencias en la función que debe sintetizar el módulo predictor a fin de reducir su complejidad. Para realizar esa selección se pueden utilizar los procedimientos espectrales que se han descrito y que, además, proporcionan de manera directa la forma de realizar la síntesis del módulo monitor de patrones.

La investigación realizada ha abierto nuevas perspectivas que se van a explorar en trabajos futuros. Así, tenemos previsto acometer las siguientes tareas:

a) Elaborar un simulador que permita aplicar al test de circuitos CMOS la técnica de selección de patrones de test basada en los cambios de coeficientes espectrales que ocasionan las faltas de anclaje de transistor. El procedimiento sería análogo al que se ha presentado aquí para los circuitos descritos al nivel de puertas lógicas con defectos modelables mediante faltas de anclaje.

b) Modificar la estructura de un LFSR para poder añadir a una secuencia generada por éste los patrones de test determinados a través de la selección de coeficientes espectrales. Como se ha mostrado, esta posibilidad puede suponer una reducción de la longitud del test en el caso de autotest integrado (BIST).

c) Obtener modelos de error para diversas estructuras de circuito mediante el procedimiento basado en la selección de muestras de defectos y combinaciones de entradas que se ha descrito aquí. Para ello, se necesita disponer de programas para la simulación de circuitos con distintos modelos de defectos, de manera que se pueda seleccionar el más adecuado. En ese sentido se dispone de un simulador que permite inyectar faltas de diferente tipo que modelan algunos de los defectos de las PLAs.

d) Realizar un estudio experimental que establezca la relación entre la complejidad del módulo predictor y el número de patrones de test que se seleccionan para distintas estructuras de circuito, teniendo en cuenta las herramientas de minimización disponibles. Con ello se pretende valorar la utilidad de realizar una selección de patrones de test o realizar la síntesis del módulo predictor para cada una de las entradas al circuito.

e) Explorar la utilización de redes neuronales para implementar procedimientos eficientes en la determinación de la función de extracción que minimiza la probabilidad de enmascaramiento.

Con la enumeración de las principales aportaciones y de las cuestiones planteadas para futuras investigaciones concluye esta Memoria que, atendiendo a las aportaciones que realiza y a las perspectivas que abre, consideramos contribuye de manera significativa al test de circuitos digitales en particular y al diseño de sistemas electrónicos digitales en general.

APENDICE I. EVALUACION DEL ENMASCARAMIENTO EN UN LFSR.

Este apéndice se dedica a estudiar la posibilidad de enmasacaramiento en LFSRs de entradas múltiples, haciendo uso de la teoría de códigos lineales.

Si el circuito a testear implementa N funciones, el número de combinaciones posibles a la salida es $q=2^N$. Puesto que q es potencia de un número primo, se puede contruir un cuerpo de Galois de q elementos, $GF(q)$, que es extensión de $GF(2)$. Cada secuencia de respuestas, $F(Y^1), F(Y^2), \dots, F(Y^L)$, a los patrones de test se puede representar mediante una secuencia de elementos de $GF(q)$, (c_1, c_2, \dots, c_L) , donde $c_i \in GF(q)$ es la N -tupla $(f_1(Y^i), f_2(Y^i), \dots, f_N(Y^i))$. Además, dada una secuencia de elementos de $GF(q)$, (c_1, \dots, c_L) , se puede definir a partir de ella el polinomio de grado $L-1$ sobre $GF(q)$, $c(x) = c_1 + c_2x + \dots + c_Lx^{L-1}$, donde la suma y el producto están definidos sobre $GF(q)$.

Por otra parte, dada una secuencia de salidas del circuito a testear, c_1, c_2, \dots, c_L , también se les puede asociar una representación vectorial mediante una L -tupla de elementos de $GF(q)$, (c_1, \dots, c_L) , que puede verse como una palabra de cierto código. En adelante, la representación vectorial y la representación polinómica de una secuencia de salidas del circuito a testear se utilizarán indistintamente. De esta manera, tendrá sentido decir que un polinomio $p(x)$ divide a una palabra del código C : se hace referencia a que el polinomio asociado a una secuencia de salidas cuya representación vectorial pertenece a un código C es divisible por $p(x)$. También se podrá hablar de que un polinomio $p(x)$ sea o no una palabra de cierto código C .

Un polinomio, $g(x)$, que divida a toda palabra de un código C se llama polinomio generador de C . Si el grado de $g(x)$ es $L-k$ y el número de elementos de una palabra del código es L , el número de símbolos de información en cada palabra es k . Si, además, el polinomio $g(x)$ divide a x^L+1 , se dice que el código C es un código cíclico de longitud L .

El resto, $r(x)$, que se obtiene al dividir una palabra $p(x)$ por el polinomio $g(x)$ generador de C recibe el nombre de síndrome en teoría de códigos. Aunque reciban el mismo nombre, no hay que confundir este síndrome con el número de unos de una función, definido en la sección 1.4. Si el grado de $g(x)$ es $L-k$, el grado de $r(x)$ será $L-k-1$ o menor.

El circuito compresor de salidas mediante división polinómica es un registro de desplazamiento de múltiples entradas con realimentación lineal (MI/LFSR, según sus siglas en inglés). El polinomio, $g(x)$, que implementa dicho registro viene definido por las realimentaciones y tiene la forma $g(x)=x+\alpha$, con $\alpha \in GF(2^N)$. Así, dada una secuencia de entradas al MI/LFSR (con todos sus biestables inicialmente a 0), el estado final del mismo corresponde a la signatura obtenida al dividir dicha secuencia por el polinomio $x+\alpha$.

Un elemento de $GF(2^N)$ se puede expresar como un polinomio de grado N sobre $GF(2)$. Además, puesto que si $g(x)=x+\alpha$ es un polinomio primitivo sobre $GF(2)$, α es un elemento primitivo de $GF(2^N)$ [LIN83, PET72], se puede considerar indistintamente que el polinomio $g(x)$ está definido sobre $GF(2)$ o sobre $GF(2^N)$ [PRA90]. Como ejemplo se tiene el MI/LFSR de la figura I.1. El polinomio del circuito es x^2+x+1 , si α es un elemento primitivo de $GF(2^2)$ definido por x^2+x+1 al dividir una secuencia de entrada cualquiera por $x+\alpha$, el resto coincide con el estado del MI/LFSR (suponiendo inicialmente todos los biestables a 0).

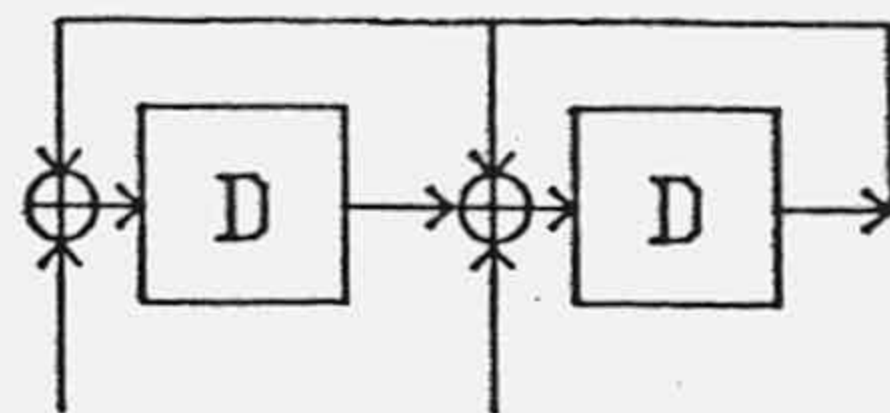


Figura I.1 Circuito LFSR de varias entradas asociado al polinomio x^2+x+1 , definido sobre el cuerpo de Galois.

Dado un MI/LFSR que implementa el polinomio $g(x)$, si $c(x)$ es la secuencia de entrada al MI/LFSR obtenida al aplicar un test a un circuito que

no presenta defectos y $c'(x)$ es dicha secuencia en el caso de que el circuito presente un defecto detectado por el test, existirá enmascaramiento si se verifica que $c(x)+c'(x)$ es una palabra del código generado por $g(x)$. Para verlo basta tener en cuenta que si existe enmascaramiento se obtendrá el mismo resto para la secuencia $c(x)$ y para la secuencia $c'(x)$:

$$\begin{aligned}c(x) &= g(x)a(x) + r(x) \\c'(x) &= g(x)b(x) + r(x)\end{aligned}$$

Por lo que, como se había indicado:

$$c(x) + c'(x) = g(x)[a(x) + b(x)]$$

Por otra parte, como $c(x)+c'(x)$ es precisamente la representación polinómica de la secuencia de errores que se obtienen al aplicar el test, $e(x) = E^1 + E^2x + \dots + E^Lx^{L-1} = c(x) + c'(x)$, se puede afirmar que existirá enmascaramiento si $e(x)$ es una palabra del código generado por $g(x)$. Así queda demostrado el siguiente teorema [PRA90]:

Teorema I.1: Un error $e(x)$ origina enmascaramiento en un MI/LFSR si y sólo si pertenece al código C , generado por el polinomio $g(x) = x + \alpha$, donde α es la raíz primitiva del correspondiente polinomio de realimentación sobre $GF(2)$.

Como consecuencia de lo anterior, si el polinomio generador es $g(x) = x + \alpha$ cualquier error $e(x)$ que cause enmascaramiento verifica que $e(\alpha) = 0$.

A continuación se establece una expresión de la probabilidad de enmascaramiento que utiliza un modelo de errores de canal q -ario, donde $q = 2^n$ y n es el número de salidas del circuito que se testea (figura 6.2). También se supondrá que la probabilidad de que se produzca un error, p , es igual para todas las combinaciones de salida, y que todos los patrones de error son equiprobables. De esta forma, la probabilidad de enmascaramiento para una secuencia de L patrones de test viene dada por la expresión:

$$p_A = \sum_{j=1}^L E(j) (1-p)^{L-j} (p/(q-1))^j \quad (I.1)$$

donde $E(j)$ es el número de patrones de error $e(x) = E^1 + E^2x + \dots + E^Lx^{L-1}$ con j coeficientes $E^k \neq 0$, $k = 1, \dots, L$, que producen enmascaramiento en la secuencia de L patrones de test aplicada. Por el teorema I.1, $E(1)$ es el número de palabras

de peso 1 del código generado por $g(x)=x+\alpha$. En el caso de que el polinomio generado por $g(x)$ sea un código separable de distancia máxima 2, el valor de $E(j)$ viene dado por [PRA90]:

$$E(j) = \binom{q-1}{1} \sum_{i=0}^{j-2} (-1)^i (q^{j-i-1} - 1) \binom{j}{i} \quad (\text{I.2})$$

y sustituyendo (I.1) en (I.2) y evaluando la expresión resultante se obtiene [PRA90] la expresión de la probabilidad de enmascaramiento para este caso:

$$p_A = 2^{-N} [(1-2^N(1-p))^L + (2^N-1)(1-2^N p/(2^N-1))^L] \quad (\text{I.3})$$

APENDICE II. PROCEDIMIENTO DE SINTESIS AND-EXOR MULTIFUNCIONAL.

En este apéndice se presenta un procedimiento la síntesis multifuncional mediante PLAs del tipo AND-EXOR.

II.1. Representación de funciones booleanas como suma módulo-2 de productos con polaridad mixta.

Se ha visto que es posible representar una función a partir de los coeficientes de su expansión en términos de las funciones de Tamari de F_2^m :

$$f(y_0, y_1, \dots, y_{m-1}) = \sum_{\alpha=0}^{t} r_f(\alpha_0, \dots, \alpha_{m-1}) y_0^{\alpha_0} \dots y_{m-1}^{\alpha_{m-1}}$$

$$\alpha = \alpha_0 + 2\alpha_1 + \dots + 2^{m-1}\alpha_{m-1}; \alpha_i \in \{0, 1\}; t = 2^m - 1 \quad (\text{II.1})$$

Los coeficientes del espectro de Tamari de la función, $r_f(\alpha)$ ($\alpha=0, \dots, t$), constituyen la representación de la función en la forma canónica de Reed-Muller. En realidad la expresión (II.1) es un caso particular del tipo de representaciones que surgen a partir de la demostración [DAR89a, GRE86, 90] de que, dado cualquier elemento $(\beta_0, \beta_1, \dots, \beta_{m-1}) \in C_2^m$ que se notará como $\beta = \beta_0 + 2\beta_1 + \dots + 2^{m-1}\beta_{m-1}$, toda función $f \in F_2^m$ se puede expresar como

$$f(y_0, y_1, \dots, y_{m-1}) = \sum_{\alpha=0}^{t} r_f^{\beta}(\alpha_0, \dots, \alpha_{m-1}) y_0(\beta_0)^{\alpha_0} \dots y_{m-1}(\beta_{m-1})^{\alpha_{m-1}} \quad (\text{II.2})$$

$$\alpha = \alpha_0 + 2\alpha_1 + \dots + 2^{m-1}\alpha_{m-1}; \alpha_i \in \{0, 1\}; t = 2^m - 1$$

$$y_j(\beta_j) = y_j \quad \text{si } \beta_j = 0, \text{ e } y_j(\beta_j) = y_j' \quad \text{si } \beta_j = 1$$

Esta es la representación de Reed-Muller con polaridad fija [GRE90, SAS90]. El número β es la polaridad de la representación y da información de qué variables aparecen complementadas en la representación de la función. La expresión (II.1) corresponde a la representación de polaridad cero ($\beta=0$) de la función, en la que ninguna variable aparece complementada. Así pues, la transformada de Tamari en F_2^m está constituida por los coeficientes del desarrollo de Reed-Muller de polaridad cero de una función binaria.

Si se pretende realizar un conjunto de funciones para las que se dispone de su representación en términos de la funciones de Tamari se puede utilizar la estructura de la figura II.1. Si se utiliza una representación con polaridad fija distinta de cero se puede usar la estructura de matriz lógica programable AND-EXOR de la figura II.2, que en ciertos casos permite reducir el espacio necesario para integrar un conjunto de funciones lógicas, en relación al que se precisaría en una matriz lógica programable del tipo AND-OR como la de la figura II.3 [SAS90]. También se podría utilizar una estructura de matriz lógica programable AND-EXOR como la de la figura II.4, en la que cada variable puede aparecer complementada y sin complementar. Una representación de este tipo de una función de F_2^m recibe el nombre de representación de Reed-Muller de polaridad mixta [GRE90, SAS90].

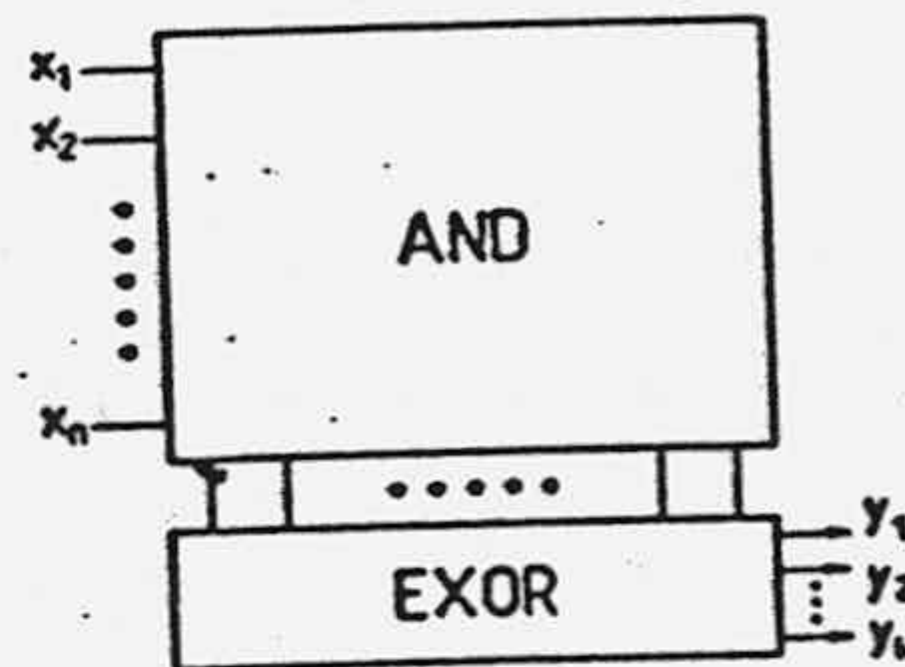


Figura II.1. Estructura PLA AND-EXOR de polaridad cero.

El interés por las estructuras de matrices programables del tipo AND-EXOR surge a partir de una serie de conjeturas hechas hace algunos años que indicaban que estas estructuras permitían una síntesis multifuncional más simple [EVE67, MUK70]. Los resultados de algunos trabajos recientes [SAS86, 90] confirman estas conjeturas. En la Tabla II.1 [SAS90] se muestran los límites superiores de los productos que se necesitan para sintetizar las funciones de n variables con distintas estructuras de PLAs. Otra razón para el interés en estas estructuras está en la dificultad del test de PLAs AND-OR dado que el conector OR tiene más bajo poder separador que el EXOR [SAS90], considerándose menos complejo el test de estructuras AND-EXOR [RED72].

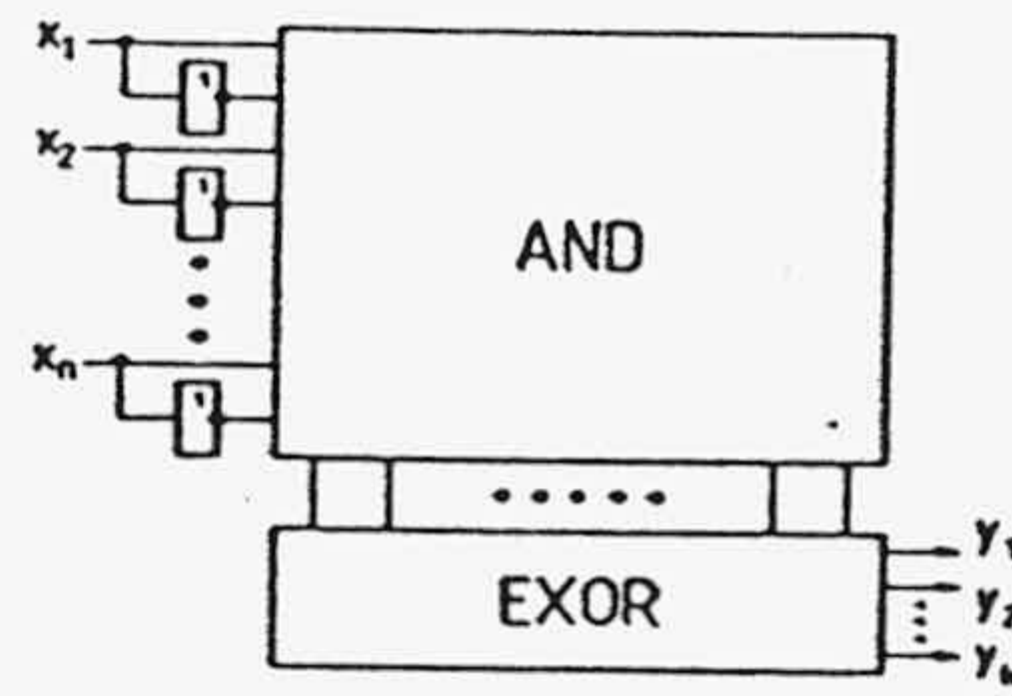


Figura II.2. Estructura de PLA AND-EXOR de polaridad fija.

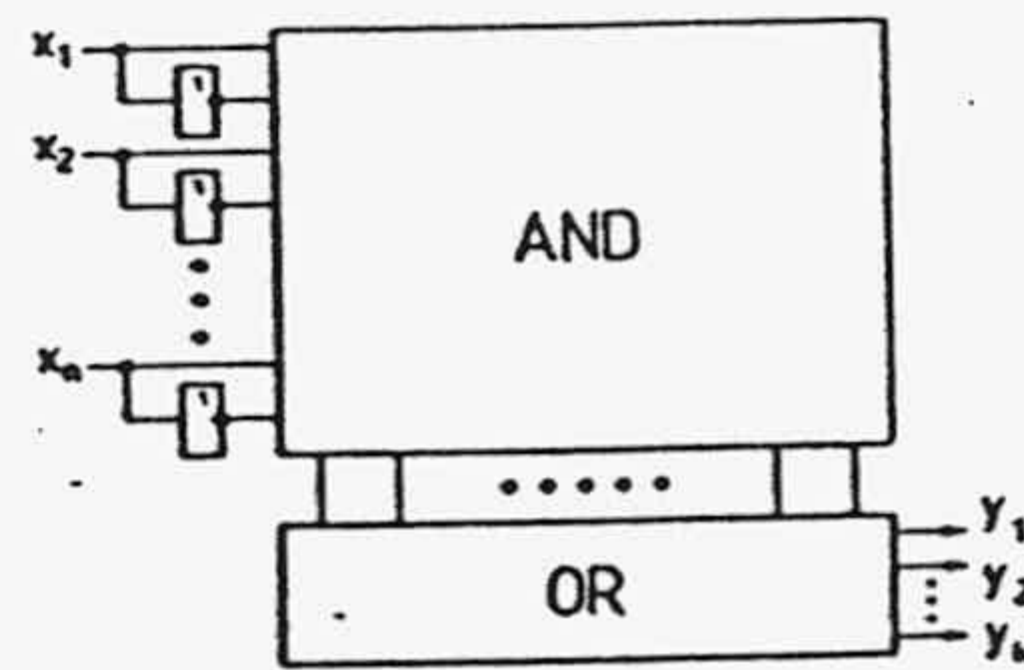


Figura II.3. Estructura de PLA AND-OR.

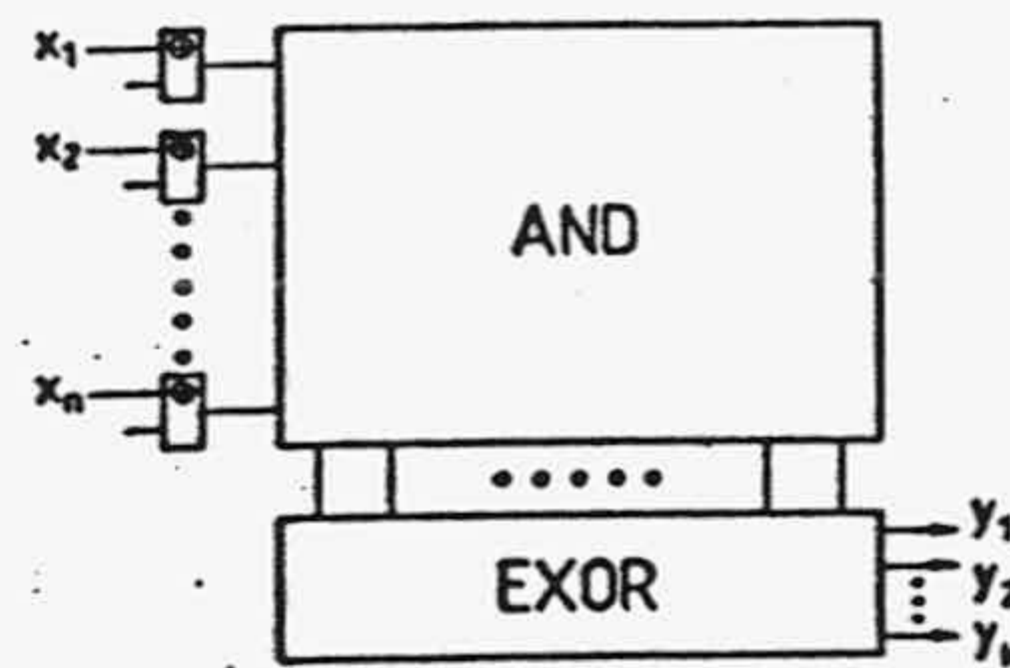


Figura II.4. Estructura de PLA AND-EXOR de polaridad mixta.

Tabla II.1. Número de líneas necesarias en distintas estructuras de PLAs.

	PLA with EXOR array	Standard PLA	PLA with two bit decoders
Arbitrary Functions	$(3/4) \cdot 2^{n-1}$	2^{n-1}	$(1/2) \cdot 2^{n-1}$
Symmetric Functions	$2 \cdot 3^{r-1}$	2^{n-1}	3^{r-1}
Parity Functions	n	2^{n-1}	2^{r-1}
n-bit Adders	$2^{n+1}-1$	$6 \cdot 2^n - 4n - 5$	$n^2 + 1$

En este apéndice se describe un procedimiento rápido para obtener una representación mínima de un conjunto de funciones en términos de sumas EXOR

de productos, con polaridades fijas o mixtas. El procedimiento proporciona un método alternativo para la minimización de funciones expresadas en términos de sumas EXOR de productos. Si se usa junto con otras herramientas para la síntesis con estructuras AND-OR, como ESPRESSO, permite elegir entre la mejor solución entre una PLA AND-OR o una PLA AND-EXOR.

II.2. Síntesis mínima AND-EXOR con polaridad mixta.

En este apartado se describe un procedimiento rápido para la síntesis de una función booleana mediante sumas módulo-2 de productos de variables complementadas y sin complementar. El procedimiento permite obtener la síntesis mínima en el sentido de que necesitará el menor número de puertas AND y por consiguiente el menor número de puertas EXOR si éstas tienen un número limitado de entradas. De esta manera se consigue que una PLA AND-EXOR tenga el menor tamaño al reducir el número de líneas producto y de puertas EXOR. Para describir el procedimiento, considérese la función de m variables $F(y_1, y_2, \dots, y_m)$. El número de productos formados con las combinaciones posibles de las m variables complementadas y sin complementar es 2^m : son los minterms de la función. La función queda determinada a partir de los minterms que aparecen en su representación. Si además de estos productos se consideran aquellos en los que no aparecen algunas variables, el número de productos posibles es 3^m . Entre estos productos se encuentran los que se utilizarán en la expresión mínima de la función como suma módulo-2 de productos de polaridad mixta. Los productos se notarán con m -tuplas de dígitos ternarios (a_1, a_2, \dots, a_m) , $(a_i = 0, 1, 2)$. Si $a_i = 0$, en el producto aparece y_i' ; si $a_i = 1$, aparece y_i ; y si $a_i = 2$, y_i no aparecerá explícitamente.

Se parte de la expresión de la función en términos de sus minterms y se toma una variable cualquiera. Para fijar ideas y sin pérdida de generalidad, sea y_1 . Las restantes $m-1$ variables son $y_2^{w_2}, \dots, y_m^{w_m}$, $w_i \in \{0, 1\}$, con $y_1^1 = y_1$ e $y_1^0 = y_1'$. El producto $y_2^{w_2} y_3^{w_3} \dots y_m^{w_m}$ se notará con Y_1^w .

Se toman los minterms $y_1 Y_1^w$ e $y_1' Y_1^w$, que se designan con las m -tuplas $(1, w_2, \dots, w_m)$ y $(0, w_2, \dots, w_m)$, respectivamente. Se puede generar el producto $(2, w_2, \dots, w_m)$, en el que ha desaparecido la variable y_1 , mediante la suma módulo-2 de $y_1 Y_1^w$ y $y_1' Y_1^w$, tal como se esquematiza en la figura II.5 para el caso de tres variables ($m=3$) y para cada uno de los posibles valores de w ($w=0, \dots, 2^{m-1}-1$). Así generamos 2^{m-1} nuevos productos a partir de los 2^m minterms de partida, resultando $2^m + 2^{m-1}$ productos, es decir $3 \cdot 2^{m-1}$.

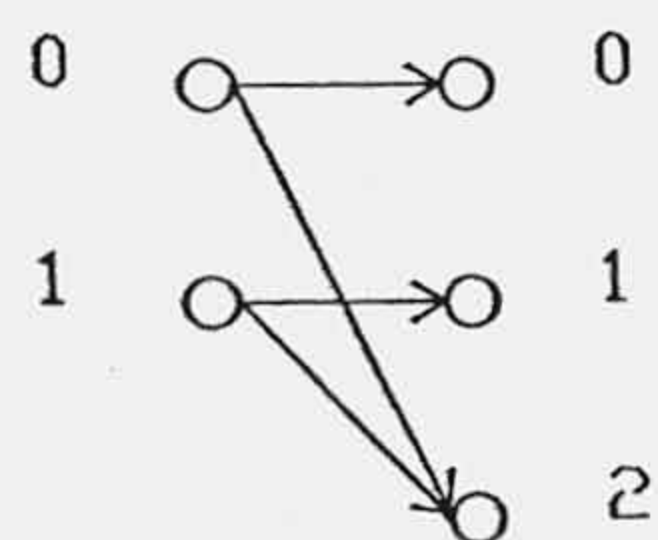


Figura II.5. Esquema del proceso para eliminar una variable.

Ahora, se repite el proceso tomando la segunda variable, y_2 . Es decir, dado el producto $(v_1, 0, w_3, \dots, w_m)$, con $v_1 = 0, 1, 2$ y $w_1 = 0, 1$, si se realiza la suma módulo-2 con el producto $(v_1, 1, w_3, \dots, w_m)$ se obtiene el producto $(v_2, 2, w_3, \dots, w_m)$, en el que no aparece la variable y_2 , y puede o no aparecer y_1 . El número de productos nuevos es $3 \cdot 2^{m-1} / 2$, por lo que el número total de productos al final de este segundo paso es $3 \cdot 2^{m-1} + 3 \cdot 2^{m-2} = 3^2 2^{m-2}$.

Si se continúa realizando estas sumas módulo-2 para cada una de las variables, se generarán los 3^m productos posibles. En el gráfico de la figura II.6 se esquematiza el proceso seguido para una función de tres variables.

Para una función de m variables, el proceso consta de m pasos y se representa mediante un grafo en el que los 3^m nudos de una misma columna corresponden a los distintos productos de variables y se numeran con el índice del producto que representan. Los valores de los nudos de la etapa siguiente se obtienen a partir de los de la anterior mediante la información que dan los arcos del grafo. Así, si dos nudos de una etapa están conectados mediante arcos con un nudo de la etapa siguiente, eso quiere decir que para obtener el valor del nuevo nudo habrá que realizar la suma módulo-2 de los nudos a él conectados. Se justifica a continuación.

Se consideran los vectores $M_i = [m_{0i} \ m_{1i}]$, $i=1, \dots, m$. En cada vector M_i se ha notado con m_{0i} al valor asociado a x_i y con m_{1i} al valor asociado a x_i en el desarrollo de una función, $f \in F_2^m$ de variables x_1, x_2, \dots, x_m de la forma:

$$\begin{aligned} f(x_1, x_2, \dots, x_m) &= x_i \cdot f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_m) + x_i \cdot f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_m) = \\ &= x_i \cdot f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_m) \oplus x_i \cdot f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_m) = x_i \cdot m_{0i} \oplus x_i \cdot m_{1i} \end{aligned}$$

A partir de m veces el producto de Kronecker de estos vectores M_i , se obtiene el vector M , cuyos componentes son los 2^m minterms de la función de m variables que se considera: la componente $\beta = 0, 1, \dots, 2^m$ corresponde al minterm $x_1^{\beta_1} \dots x_m^{\beta_m}$ con $\beta = (\beta_1, \dots, \beta_m)$ y su valor será 1 ó 0 según sea el valor de la

función en dicho minterm.

$$M = M_m \otimes M_{m-1} \otimes \dots \otimes M_2 \otimes M_1 \tag{II.3}$$

Por otra parte, se puede definir un vector asociado a cada variable, $E_i = [a_i \ b_i \ c_i]$ ($i=1, \dots, m$), representando las distintas formas en que pueden aparecer los valores m_{0i} y m_{1i} de la función $f \in F_2^m$, en las expresiones de f como suma módulo-2 para cada una de las bases posibles, es decir:

$$\{1, x_i\}, \quad \{1, x_i'\}, \quad \{x_i', x_i\} \tag{II.4}$$

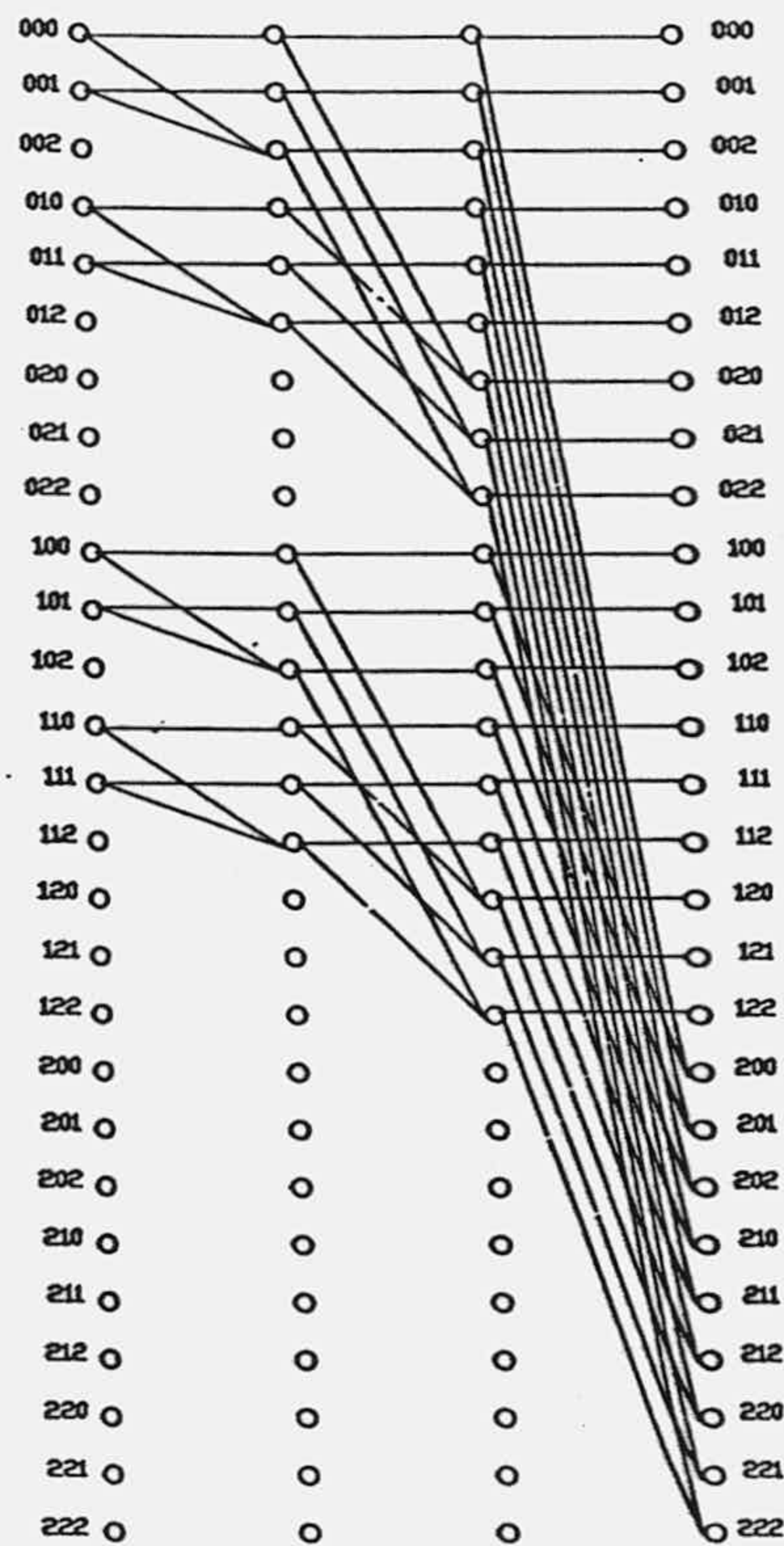


Figura II.6. Generación de los 27 productos de una función de tres variables.

Consecuentemente, como las expresiones de $f \in F_2^m$ en términos de estas bases son, respectivamente:

$$\begin{aligned} f(x_1, \dots, x_m) &= m_{0i} \oplus (m_{0i} \oplus m_{1i}) x_i \\ f(x_i) &= (m_0 \oplus m_1) \oplus m_1 x_i' \\ f(x_i) &= m_0 x_i' \oplus m_1 x_i \end{aligned} \tag{II.5}$$

Las distintas formas en las que pueden aparecer m_{0i} y m_{1i} son

$$a_i = m_{0i} \quad b_i = m_{1i}, \quad c_i = m_{0i} \oplus m_{1i} \tag{II.6}$$

Se define así una matriz que relaciona el vector E_i con el vector M_i :

$$E_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \quad M_i(\alpha)^T = P_i M_i(\alpha)^T \quad (\text{II.7})$$

Si en la representación de la función $f \in F_2^m$ como suma módulo-2 de productos se utiliza la variable x_i sin complementar, se dice que la polaridad con la que interviene x_i es 0, si se utiliza la variable x_i complementada la polaridad es 1 y, si se utiliza la variable x_i complementada y sin complementar, la polaridad es 2. Se define entonces el vector $S_i = [s_{i,0}, s_{i,1}, s_{i,2}]$, $i=1, \dots, m$, cuya componente $s_{i,j}$ indica el número de términos que tiene la representación de la función $f(x_1, \dots, x_m) \in F_2^m$ con polaridad j ($j=0,1,2$) en la variable x_i .

La relación entre el vector E_i y el vector S_i viene dada por

$$S_i = E_i \cdot C_i \quad (\text{II.8})$$

donde el producto de la matriz C_i por el vector $E_i(\alpha)$ se define sobre los reales. Según (II.6) la matriz C_i tiene la forma:

$$C_i = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad (\text{II.9})$$

El producto de Kronecker de los vectores S_i , $i=1,2,\dots,m$, permite definir el vector S , de 3^m componentes reales, s_i , $i=0,\dots,3^m-1$:

$$S = S_m \otimes S_{m-1} \otimes \dots \otimes S_0 \quad (\text{II.10})$$

donde la componente s_i es igual al número de términos producto que aparecen en la representación de la función representada con polaridad $i = w_1 + 3w_2 + \dots + 3^{m-1}w_m$, siendo $w_i \in \{0,1,2\}$ la polaridad con la que interviene la variable x_i . Así,

$$S = (E_m \cdot C_1) \otimes \dots \otimes (E_1 \cdot C_1) = (E_1 \otimes \dots \otimes E_m) (C_1 \otimes \dots \otimes C_1) \quad (\text{II.11})$$

El producto de Kronecker de los vectores E_i , $i=1,\dots,m$, define el llamado vector extendido, formado por las 3^m componentes $e_i = \{0,1\}$, $i=0,1,\dots,3^m-1$.

$$E = E_m \otimes E_{m-1} \otimes \dots \otimes E_2 \otimes E_1 \quad (\text{II.12})$$

Teniendo en cuenta que E y M se construyen a partir del producto de Kronecker de los E_i y M_i respectivamente, según las expresiones (II.5) y (II.12), y utilizando las propiedades del producto de Kronecker, se tiene que

$$E = M \cdot P_m \quad \text{con} \quad P_m = P_1 \otimes \dots \otimes P_m \dots \otimes P_1 \quad (\text{II.13})$$

La transformación que describe esta matriz corresponde a la que representa el grafo descrito anteriormente, en el que el valor de cada nudo en cada una de las etapas se obtiene mediante la suma módulo-2 de los valores de los nudos que confluyen en él. Considerando funciones booleanas sin indiferencias, los nudos de la primera etapa se inician como sigue:

- a) Los nudos correspondientes a minterms donde la función se haga cero se inician a cero.
- b) Los nudos correspondientes a minterms donde la función se haga uno se inician a uno.
- c) El resto de los nudos de la primera etapa se inician a cualquier valor entero distinto de uno o cero.

Al terminar la última etapa del proceso, los nudos serán uno o cero según intervenga o no el producto correspondiente en alguna de las expresiones AND-EXOR de la función con polaridad mixta.

Como consecuencia de todo esto, para determinar la polaridad para la que es mínimo el número de términos y determinar la representación de la función en dicha polaridad habrá que completar los siguientes pasos:

- a) Obtener el vector extendido a partir de la representación de la función en términos de sus minterms.
- b) Obtener el vector S multiplicando el vector E por la matriz C y elegir la componente de menor valor, s_1 .
- c) Seleccionar los elementos del vector extendido E que tengan índices iguales a los de las columnas para las que la fila i -ésima de la matriz C tiene unos. En total serán 2^m elementos con los que se forma el vector de representación $V_1 \in C_2^t$ ($t=2^m$).
- d) Considerar el vector con componentes iguales a los términos de la representación con polaridad i , R_1 . Si la componente j -ésima del vector V_1 es uno, en la representación de la función aparecerá el término que corresponde a la componente j -ésima en el vector R_1 .

Como ejemplo, se puede considerar la función

$$f(x_1, x_2, x_3) = x_1 x_2 \oplus x_3 \oplus x_1 x_3$$

para la que el vector M es

$$M = [0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1]$$

y, aplicando la matriz P_3 se obtiene el vector E correspondiente:

$$E=[0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0]$$

Se multiplica el vector E por la matriz $C(3)=C_1 \otimes C_1 \otimes C_1$ para determinar el vector S_3

$$S_3=[3\ 3\ 2\ 4\ 5\ 3\ 5\ 4\ 3\ 5\ 4\ 3\ 4\ 4\ 4\ 7\ 4\ 5\ 4\ 5\ 3\ 4\ 7\ 5\ 4\ 4\ 4]$$

y puesto que la tercera componente de este vector es la menor, la polaridad para la que se minimiza el número de términos producto es 2, que expresada en ternario corresponde a la combinación $(0,0,2)$. Por consiguiente, x_3 y x_2 aparecerán sin complementar y x_1 aparece complementada y sin complementar. A partir de la fila tercera de la matriz S_3 se determina el vector R_2 , que es igual a

$$R_2=[1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0]$$

y el vector V_2 es:

$$V_2=[e_1\ e_2\ e_6\ e_7\ e_{18}\ e_{19}\ e_{24}\ e_{25}]=[0\ 0\ 0\ 1\ 1\ 0\ 0\ 0]$$

Puesto que la polaridad es $(0,0,2)$, los productos que intervienen se obtienen a partir del producto de Kronecker [GRE90, HUR85]

$$[1\ x_3] \otimes [1\ x_2] \otimes [x_1' \ x_1] = [x_1' \ x_1 \ x_2 x_1' \ x_2 x_1 \ x_3 x_1' \ x_3 x_1 \ x_3 x_2 x_1' \ x_3 x_2 x_1]$$

es decir, son $x_2 x_1$ y $x_3 x_1'$ y la función se expresará como:

$$f(x_3, x_2, x_1) = x_2 x_1 \otimes x_3 x_1'$$

En el caso de que la función presente indiferencias, el proceso que se sigue es el mismo salvo que no se asigna 0 ó 1 a los valores iniciales de los nudos que corresponden a minterms indiferentes. A lo largo del proceso de obtención se van dando valores a las indiferencias a medida que se van realizando las sumas módulo-2 que las involucran. Para ello se puede utilizar el algoritmo siguiente:

(A) Supóngase el paso de cálculo i -ésimo en el que hay que sumar el contenido del nudo $(\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_m)$ con el del nudo $(\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_m)$ para obtener el del $(\alpha_1, \dots, \alpha_{i-1}, 2, \alpha_{i+1}, \dots, \alpha_m)$. Esos contenidos se notarán $N(0)$, $N(1)$ y $N(2)$, respectivamente, y pueden ser 0, 1 ó X (indiferencia).

(A.1) Si $N(0)=N(1)=X$ hacer $N(2)=X$.



(A.2) Si $N(0)=0$ ($N(1)=0$) y $N(1)=X$ ($N(0)=X$), hacer $N(1)=0$ ($N(0)=0$) y, consecuentemente $N(2)=0$.

(A.3) Si $N(0)=1$ ($N(1)=1$) y $N(1)=X$ ($N(0)=X$), hacer $N(1)=N(0)$ ($N(0)=N(1)$) y, consecuentemente $N(2)=0$.

(B) Al terminar el cálculo de E, todas las indiferencias que aparezcan se igualan a cero.

El objetivo que persigue este algoritmo es que al final el número de ceros en el vector E sea el mayor posible a fin de que los valores que tomen las componentes del vector S_m tengan el menor valor posible. Teniendo esto en cuenta, la justificación de (A.1), (A.2) y (B) resulta evidente.

En el caso de (A.3), existen dos posibilidades: hacer que $N(1)$ ($N(0)$) sea 0 con lo que $N(2)$ sería 1 o bien hacer $N(1)=N(0)$ y $N(2)=0$. Se ha elegido la segunda, debido a que así se introduce un cero y a que nunca aparecerán en una suma EXOR mínima dos productos que sólo difieran en una variable ya que en ese caso se podría reducir el número de términos inmediatamente, sustituyéndolos.

Así, si se considera la función del ejemplo anterior, pero con indiferencias tales que el vector M sea:

$$M=[X \ X \ 0 \ 1 \ 1 \ X \ 1 \ 1]$$

aplicando el algoritmo anterior, el vector extendido es

$$E=[0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]$$

que, como se puede comprobar, tiene 12 unos en lugar de los seis del vector E de la función sin adyacencias. Al aplicar ahora la matriz P_3 se obtiene el siguiente vector R_3 :

$$R_3=[3 \ 3 \ 2 \ 3 \ 3 \ 2 \ 6 \ 6 \ 5 \ 3 \ 2 \ 3 \ 3 \ 2 \ 3 \ 6 \ 4 \ 6 \ 2 \ 3 \ 3 \ 2 \ 3 \ 3 \ 4 \ 6 \ 7]$$

existiendo varias polaridades para las que el número de términos es 2. Entre ellas, está la correspondiente al caso en el que no existían indiferencias.

Si tomamos la componente sexta del vector R_3 , se obtiene

$$V_5=[e_3 \ e_4 \ e_6 \ e_7 \ e_{21} \ e_{22} \ e_{24} \ e_{25}]=[0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]$$

y la expresión de la función con polaridad 0 para x_3 , polaridad 1 para x_2 y polaridad 2 para x_1 es

$$f(x_3, x_2, x_1) = x_1 \oplus x_1' x_3$$

que correspondería al vector $M=[0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1]$.

II.3. Síntesis multifuncional mínima AND-EXOR con polaridad mixta.

Una vez expuesto el procedimiento para sintetizar una función booleana como sumas módulo-2 de productos de funciones con distinta polaridad, en este apartado se presenta una modificación de dicho procedimiento para la síntesis multifuncional de funciones con o sin indiferencias. Para ello, se supone que se desea sintetizar las s funciones de F_2^m $f_1(x_1, \dots, x_m), \dots, f_s(x_1, \dots, x_m)$. El vector E para la función f_i ($i=1, \dots, s$) se notará $E(i)$.

A continuación se presentan dos procedimientos: uno aplicable si las funciones no tienen indiferencias y otro para el caso en el que las tengan.

Caso 1. Funciones sin indiferencias.

La obtención del correspondiente vector $E(i)$ para cada una de las funciones se hace como en el caso de síntesis de una función. A partir de ahí, el procedimiento de síntesis multifuncional continuaría con los siguientes pasos:

- (1) Obtener un nuevo vector extendido E para las s funciones mediante la expresión

$$E_i = E(1)_i \cup E(2)_i \cup \dots \cup E(s)_i$$

$$i=0, \dots, 3^m-1$$

donde E_i y $E(j)_i$ indican, respectivamente, la componente i -ésima del vector E y del vector $E(j)$.

- (2) Obtener el vector S a partir del vector E , aplicando la expresión

$$S = E \cdot C(m)$$

- (3) Determinar la polaridad de la representación a partir del índice de la menor componente del vector S .
- (4) Obtener la representación de cada función f_i ($i=1, \dots, s$) con la polaridad seleccionada en el apartado (3) y mediante el uso del correspondiente vector extendido E_i ($i=1, \dots, s$).

En este algoritmo, el paso (1) persigue la generación de un vector que contenga los términos que intervienen debido a todas las funciones. Sería el vector E correspondiente a la operación OR de todas las funciones. En los pasos (2) y (3) se busca la polaridad que permite realizar esa función con el mínimo número de términos, y en el paso (4) se determina qué términos corresponden a cada función, con la polaridad seleccionada.

Caso 2. Funciones con indiferencias.

Para esta situación hay varias actuaciones posibles:

(a) Por una parte se podría:

(a.1) Aplicar el procedimiento de obtención de los vectores $E(i)$ para cada función con indiferencias como si fueran independientes, y

(a.2) Aplicar el procedimiento de síntesis multifuncional para funciones sin indiferencias del caso 1.

(b) Por otra, se trataría de asignar valores a las indiferencias de las funciones de manera que el vector E que resultara al realizar la unión de todos los $E(i)$, $i=1, \dots, s$, tuviera el mínimo número de unos posible. Para conseguir esto, el procedimiento a aplicar sería el siguiente:

(b.1) Supóngase el paso i -ésimo donde hay que sumar el contenido del nudo $(\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_m)$ con el del nudo $(\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_m)$ para obtener el del $(\alpha_1, \dots, \alpha_{i-1}, 2, \alpha_{i+1}, \dots, \alpha_m)$ en cada una de las funciones. Para la función f_i ($i=1, \dots, s$), esos contenidos se notarán $N_i(0)$, $N_i(1)$ y $N_i(2)$, respectivamente y, como vimos, pueden ser 0, 1 ó X.

(A) Si $N_i(0)=0$ ($N_i(1)=0$) y $N_i(1)=X$ ($N_i(0)=X$), hacer $N_i(1)=0$ ($N_i(0)=0$) y, consecuentemente $N_i(2)=0$.

(B) Si $N_i(0)=1$ ($N_i(1)=1$) y $N_i(1)=X$ ($N_i(0)=X$), hacer $N_i(1)=N_i(0)$ ($N_i(0)=N_i(1)$) y, consecuentemente $N_i(2)=0$.

(C) Si $N_i(0)=N_i(1)=X$:

- Hacer $N_i(0)=0$, $N_i(1)=0$ y $N_i(2)=0$, si hay algún $j=1, \dots, s$ ($j \neq i$) para el cual se verifica que $N_j(0)=N_j(1)=0$ antes o después de aplicar (A).

- En caso contrario, hacer $N_i(0)=1$, $N_i(1)=1$ y $N_i(2)=1$, si hay algún $j=1, \dots, s$ ($j \neq i$) para el cual se verifica que $N_j(0)=N_j(1)=1$ antes o después de aplicar (B).

- Si no se verifica ninguno de los casos anteriores, hacer $N_i(2)=X$.

(b.2) Al terminar el cálculo de cada $E(i)$, $i=1, \dots, s$, todas las indiferencias que aparezcan se igualan a cero.

(3) Se repiten los pasos (1), (2), (3) y (4) aplicados en el caso 1.

APENDICE III. PROPIEDADES DE LOS COEFICIENTES ESPECTRALES BINARIOS.

En este apéndice se demuestran una serie de propiedades de la transformada de Tamari para funciones binarias, $f \in F_2^m$. Estas propiedades, con el mismo número, se enunciaron en el capítulo 3.

Propiedad 3.1: Para todo $i=(i_1, \dots, i_m)$ con $i_j \in C_2$ se verifica que

$$r_f(i) = \sum'_{Y=0}^N T_Y(i) f(Y) = \sum'_{Y \subseteq i} f(Y)$$

donde $Y=(y_1, \dots, y_m)$ con $y_j \in C_2$ e $Y \subseteq i$ si y sólo si $y_j \leq i_j \quad \forall j=1, \dots, m$.

Demostración: Como $f(Y) = \sum'_{i=0}^r r_f(i) T_i(Y)$, y utilizando la propiedad 3.1, se tiene que

$$\begin{aligned} \sum'_{Y=0}^r f(Y) T_Y(i) &= \sum'_{Y=0}^r (\sum'_{k=0}^r r_f(k) T_k(Y)) = \\ &= \sum'_{k=0}^r r_f(k) (\sum'_{Y=0}^r T_k(Y) t_Y(i)) = \sum'_{k=0}^r r_f(k) \delta_{k,i} = r_f(i) \end{aligned}$$

Además, como $T_Y(i) = i_1^{y_1} \dots i_m^{y_m}$, se verifica que $T_Y(i) = 1$ si $y_j \leq i_j \quad (j=1, \dots, m)$, y $T_Y(i) = 0$ en otro caso, y por consiguiente

$$r_f(i) = \sum'_{Y=0}^r f(Y) T_Y(i) = \sum'_{Y \subseteq i} f(Y) \quad \blacksquare$$

Propiedad 3.2: Para cualesquiera $i=(i_1, \dots, i_m)$ e $Y=(y_1, \dots, y_m)$ con $y_j, i_j \in C_2 \quad (j=1, \dots, m)$, se verifica que $\sum'_{i=0}^r T_j(i) T_i(Y) = \delta_{j,Y}$, donde $\delta_{a,b} = 1$ si $a=b$ y $\delta_{a,b} = 0$ en otro caso.

Demostración: Puesto que $T(2,m) = T(2,1) \otimes \dots \otimes T(2,1)$ y, teniendo en cuenta que $T(2,1) = T(2,1)^{-1}$ (3.3), las filas de $T(2,m)$ y $T(2,m)^{-1}$ coinciden. Además, el elemento de matriz $t_{j,i}$ verifica que $T_j(i) = t_{j,i}$. Si $t'_{i,k}$ es el elemento de la fila i y la columna k de la matriz $T(2,m)^{-1}$, puesto que esta matriz coincide con $T(2,m)$, se tendrá que $T_i(k) = t'_{i,k}$. Por tanto $\sum'_{i=0}^r T_j(i) T_i(Y) = \sum'_{i=0}^r t_{j,i} t'_{i,Y} = \delta_{j,Y} \quad \blacksquare$

Propiedad 3.3: Si las funciones $f, g, h \in F_2^m$ verifican que $f(Y) = g(Y) \oplus h(Y)$, $\forall Y \in C_2^m$, se cumple que $r_f(i) = r_g(i) \oplus r_h(i)$, $\forall i = 0, 1, \dots, N$.

Demostración: Puesto que $f(Y) = \sum'_{i=0}^r r_f(i) T_i(Y)$, $g(Y) = \sum'_{i=0}^r r_g(i) T_i(Y)$, y $h(Y) = \sum'_{i=0}^r r_h(i) T_i(Y)$, se tiene que si $f(Y) = g(Y) \oplus h(Y)$,

$$\sum'_{i=0}^r r_f(i) T_i(Y) = (\sum'_{i=0}^r r_g(i) T_i(Y)) \oplus (\sum'_{i=0}^r r_h(i) T_i(Y)),$$

y agrupando términos se obtiene

$$\sum'_{i=0}^r r_f(i) T_i(Y) = \sum'_{i=0}^r (r_g(i) \oplus r_h(i)) T_i(Y).$$

Identificando coeficientes en ambos términos queda

$$r_f(i) = r_g(i) \oplus r_h(i) \quad \blacksquare$$

Propiedad 3.4: Si $f, g, h \in F_2^m$ verifican que $f(Y) = g(Y)h(Y)$, $\forall Y \in C_2^m$, entonces $r_f(i) = \sum'_{a \cup b = i} r_g(a) r_h(b)$, con $a = (a_1, a_2, \dots, a_m)$; $b = (b_1, b_2, \dots, b_m)$; $i = (i_1, i_2, \dots, i_m)$; $a_j, b_j, i_j \in C_2$ y definiéndose $a \cup b = (a_1 \cup b_1, \dots, a_m \cup b_m)$.

Demostración: Partiendo de que $f(Y) = \sum'_{i=0}^r r_f(i) T_i(Y)$,

$g(Y) = \sum'_{a=0}^r r_g(a) T_a(Y)$, y $h(Y) = \sum'_{b=0}^r r_h(b) T_b(Y)$, se tiene que

$$\begin{aligned} g(Y)h(Y) &= (\sum'_{a=0}^r r_g(a) T_a(Y)) (\sum'_{b=0}^r r_h(b) T_b(Y)) = \\ &= \sum'_{a=0}^r \sum'_{b=0}^r r_g(a) r_h(b) T_a(Y) T_b(Y) = \\ &= \sum'_{a=0}^r \sum'_{b=0}^r r_g(a) r_h(b) T_{a \cup b}(Y) = \\ &= \sum'_{i=0}^r (\sum'_{i = a \cup b} r_g(a) r_h(b)) T_i(Y), \end{aligned}$$

e identificando coeficientes $r_f(i) = \sum'_{i = a \cup b} r_g(a) r_h(b)$ ■

Propiedad 3.5: Si $f, g, h \in F_2^m$ verifican que $f(Y) = g(Y)Uh(Y)$, $\forall Y \in C_2^m$, entonces $r_f(i) = r_g(i) \oplus r_h(i) \oplus (\sum'_{a \cup b = i} r_g(a) r_h(b))$

Demostración: Puesto que $f(Y) = g(Y)Uh(Y) = g(Y) \oplus h(Y) \oplus g(Y)h(Y)$, utilizando las propiedades 3.3 y 3.4 demostradas, se tendrá que

$$r_f(i) = r_g(i) \oplus r_h(i) \oplus (\sum'_{i = a \cup b} r_g(a) r_h(b)) \quad \blacksquare$$

Propiedad 3.6: Si los espectros de $f, g, h \in F_2^m$ verifican que $r_f(i) = r_g(i) r_h(i)$, $\forall i = 0, 1, \dots, r$, se cumple que $f(Y) = \sum'_{x \cup z = y} g(X)h(Z)$, con $X, Y, Z \in C_2^m$.

Demostración: A partir de la propiedad 3.2 es $r_f(i) = \sum'_{y \subseteq i} f(Y)$,

$r_g(i) = \sum'_{x \subseteq i} g(X)$, $r_h(i) = \sum'_{z \subseteq i} h(Z)$ y, consiguientemente,

$$\begin{aligned} r_f(i) &= \sum'_{y \subseteq i} f(Y) = (\sum'_{x \subseteq i} g(X)) (\sum'_{z \subseteq i} h(Z)) = \\ &= \sum'_{x \subseteq i} \sum'_{z \subseteq i} (g(X)h(Z)) = \sum'_{y \subseteq i} (\sum'_{z \cup x = y} (g(X)h(Z))), \end{aligned}$$

es decir, identificando términos, $f(Y) = \sum'_{z \cup x = y} (g(X)h(Z))$ ■

APENDICE IV. TEST DE FALTAS DE ANCLAJE DE TRANSISTOR EN CIRCUITOS CMOS.

Este apéndice presenta la utilización de los coeficientes del espectro de Tamari para el test de las faltas de transistor anclado en circuitos CMOS. Se trata de las faltas denominadas de stuck-open y stuck-on. En los circuitos CMOS, estas faltas no se pueden modelar como faltas de anclaje al nivel de descripción de circuito lógico. No obstante, es posible realizar cierta descripción de la función realizada por el circuito CMOS mediante puertas lógicas que sí permite modelar las faltas de anclaje de los transistores como faltas de anclaje en las entradas de ese circuito [JAI85, RED88]. Como ejemplo, considérese la puerta NOR CMOS y su descripción de puertas lógicas para el modelado de faltas de transistor anclado de la figura 1.13. En esa descripción existe un subcircuito que corresponde a la función de unos que realiza la red de transistores PMOS, y se notará con $F_p(Y)$, y otro subcircuito que corresponde a la función de ceros que realizan los transistores NMOS de la puerta, y se notará con $F_n(Y)$. Es decir, $F_p(Y)=1$ ($F_n(Y)=1$) si cuando se aplica Y existe un camino de conducción entre V_{DD} (V_{SS}) y la salida y $F_p=0$ ($F_n=0$) en caso contrario. Por consiguiente, en un circuito CMOS con m entradas, estas dos funciones verifican que

$$F_p(Y) = F_n(Y) = 1 \oplus F_n(Y), \quad \forall Y \in C_2^m.$$

Si existe algún transistor anclado-abierto (stuck-open) en la red de transistores PMOS (NMOS), la falta se detectaría aplicando una combinación de entradas para la que exista camino entre V_{DD} (V_{SS}) siempre que el nudo de salida este a 0 (1). Si hay una falta de transistor anclado a conducción

(stuck-on) en la red de transistores PMOS (NMOS), la detección del defecto se realiza mediante alguna combinación de entradas para la que exista un camino de conducción en la red de transistores NMOS y PMOS al mismo tiempo. No obstante, hay que tener en cuenta que la salida para este caso depende de la relación de impedancias de los caminos en la red PMOS y NMOS.

Como $F_p(Y) = 1 \ominus F_n(Y)$, se tiene que $r_p(j) = r_n(j) \quad \forall j = 1, \dots, N$, y $r_p(0) = 1 \ominus r_n(0)$; siendo $r_p(j)$ los coeficientes espectrales de $F_p(Y)$, $r_n(j)$ los coeficientes espectrales de $F_n(Y)$, $Y = (y_1, \dots, y_m)$, $j = (j_1, \dots, j_m)$ y $N = 2^m - 1$.

Una falta de anclaje en un transistor PMOS (NMOS), a cuya base se aplica la entrada y_1 , ocasiona que se anulen algunos de los coeficientes espectrales de la función F_p (F_n) de entre aquellos cuyo índice $j = (j_1, \dots, j_1, \dots, j_n)$ es tal que $j_1 \neq 0$. Si la falta se produce en la red de transistores PMOS (NMOS) sólo se verán afectados los coeficientes de F_p (F_n), quedando invariantes los de F_n (F_p). Así, considerando el circuito de la figura IV.1, F_p y F_n son

$$F_p = (y_4 + y_1 y_3 + y_2 y_3)' = 1 \ominus y_4 \ominus y_1 y_3 \ominus y_2 y_3 \ominus y_1 y_2 y_3 \ominus y_1 y_3 y_4 \ominus y_2 y_3 y_4 \ominus y_1 y_2 y_3 y_4$$

$$F_n = y_4 + y_1 y_3 + y_2 y_3 = y_4 \oplus y_1 y_3 \oplus y_2 y_3 \oplus y_1 y_2 y_3 \oplus y_1 y_3 y_4 \oplus y_2 y_3 y_4 \oplus y_1 y_2 y_3 y_4$$

con lo que los coeficientes espectrales distintos de cero para F_p y F_n son $r_p(0000)$, $r_p(0001) = r_n(0001)$, $r_p(1010) = r_n(1010)$, $r_p(0110) = r_n(0110)$, $r_p(1110) = r_n(1110)$, $r_p(1011) = r_n(1011)$, $r_p(0111) = r_n(0111)$ y $r_p(1111) = r_n(1111)$.

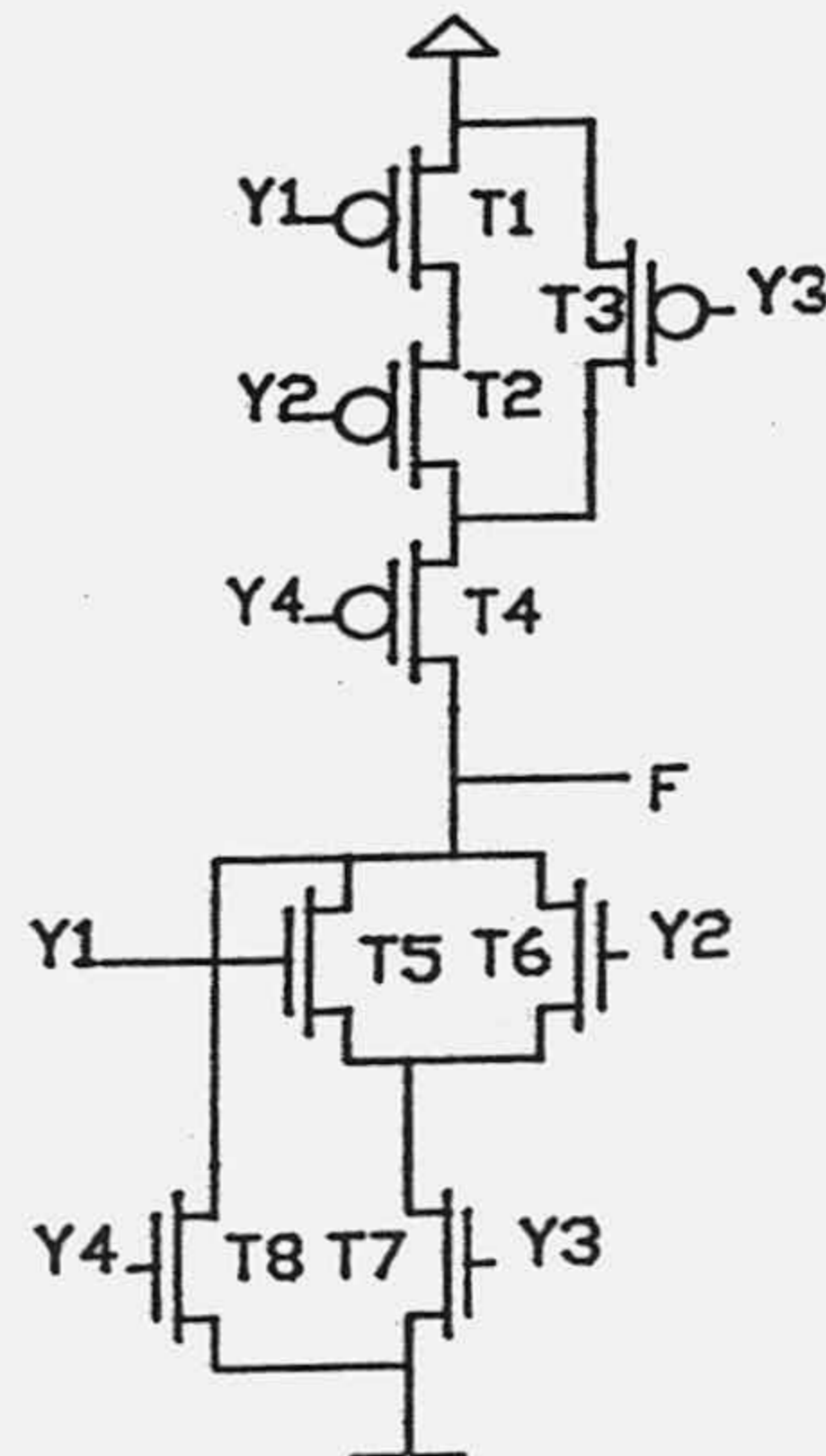


Figura IV.1: Circuito CMOS.

Supóngase una falta de anclaje abierto (stuck-open) del transistor T6. En ese caso se suprime el camino de conducción a tierra que existía cuando $y_2 = 1$ y $F_n = y_4 + y_1 y_2$, por lo que todos los coeficientes espectrales $r_n(j_1, j_2, j_3, j_4)$ con $j_2 \neq 0$ se anularán, es decir, $r_n(0110)$, $r_n(1110)$, $r_n(0111)$

y $r_n(1111)$. Los coeficientes espectrales de F_p se mantienen invariantes. Si, por ejemplo, se produjese una falta de anclaje a corto (stuck-on) en el transistor T8 se anularían todos los coeficientes espectrales de la función F_n , $r_n(j_1, j_2, j_3, j_4)$, en los que $j_4 \neq 0$. Además todos aquellos coeficientes de F_n para los que $j_4 = 0$ se transforman, obteniéndose los coeficientes

$$r'_n(j_1, j_2, j_3, 0) = r_n(j_1, j_2, j_3, 0) \oplus r_n(j_1, j_2, j_3, 1).$$

Una vez se ha establecido qué coeficientes espectrales cambian debido a una falta de anclaje en un transistor, se trata de utilizarlos para determinar los patrones de test para detectarlos. En ese sentido, hay que tener en cuenta que si un coeficiente espectral $r_n(j)$ ($r_p(j)$), $j \neq 0$, es igual a 1, eso significa que F_n (F_p) es igual a 1 para un número impar de aquellas entradas de las que dependa ese coeficiente y, puesto que el número de entradas de las que depende un coeficiente es par, deben existir entradas para las que se hace cero. Además hay que tener en cuenta que $F_p(Y) = 1 \oplus F_n(Y)$ y que, por consiguiente $r_p(j) = r_n(j)$, $j \neq 0$.

En lo que sigue, se considerarán faltas de anclaje de transistor en la red de transistores NMOS. El análisis es análogo para PMOS. Sea una falta de anclaje en un transistor NMOS que hace que el coeficiente $r_n(j) \neq 0$, $j = (j_1, \dots, j_m)$, se anule. Las combinaciones de entrada de las que dependen $r_n(j)$ y $r_p(j)$ son aquellas $Y = (y_1, \dots, y_m)$ para las que $y_i \leq j_i$ ($i = 1, \dots, m$). Las designaremos Y_1, \dots, Y_r , con r par y las consideramos ordenadas de menor a mayor. Para esa secuencia de entradas, F_n y F_p toman el valor 0 y 1, y por consiguiente se dará al menos una de las dos situaciones que se indican:

- a) $F_n(Y_i) = 0, F_n(Y_{i+1}) = 1; F_p(Y_i) = 1, F_p(Y_{i+1}) = 1$ para algún $i = 1, \dots, r-1$;
- b) $F_n(Y_k) = 1, F_n(Y_{k+1}) = 0; F_p(Y_k) = 0, F_p(Y_{k+1}) = 1$ para algún $k = 1, \dots, r-1$.

En primer lugar, sea una falta de transistor anclado abierto, que afectaría a la función $F_n(Y)$ en algunas de las combinaciones de entradas para las que $F_n(Y) = 1$, ya que por esta falta desaparecen caminos de conducción entre salida y tierra, es decir, desaparecen unos en F_n . Supóngase un valor de la función F_n afectado en Y_k . Se pueden dar las siguientes posibilidades:

- a) $F_n(Y_{k-1}) = 1, F_n(Y_k) = 1; F_p(Y_{k-1}) = 0, F_p(Y_k) = 0$; es decir $F(Y_{k-1}) = 0, F(Y_k) = 0$;
 - b) $F_n(Y_{k-1}) = 0, F_n(Y_k) = 1; F_p(Y_{k-1}) = 1, F_p(Y_k) = 0$; es decir $F(Y_{k-1}) = 1, F(Y_k) = 0$;
- y los cambios que se ocasionarían son:

- a) $F_n(Y_{k-1}) = 1, F_n(Y_k) = 0; F_p(Y_{k-1}) = 0, F_p(Y_k) = 0$; es decir $F(Y_{k-1}) = 0, F(Y_k) = 0$;
- b) $F_n(Y_{k-1}) = 0, F_n(Y_k) = 0; F_p(Y_{k-1}) = 1, F_p(Y_k) = 0$; es decir $F(Y_{k-1}) = 1, F(Y_k) = 1$.

Por consiguiente, la falta se detecta en el caso b. Esto es, si la secuencia que se toma para los patrones de entrada de los que depende el coeficiente $r_n(j)$ es tal que los valores Y_k , para los que $F_n(Y_k)$ cambia debido

a la falta, están precedidos de patrones para los que $F_n = 0$.

Si se produce una falta de anclaje a corto en un transistor NMOS, las combinaciones Y_k para las que puede cambiar el de F_n son aquellas que verifican que $F_n(Y_k) = 0$, ya que una falta de anclaje a corto en un transistor ocasiona que exista un camino para una combinación para la que antes no existía. Es decir, aparecen unos en F_n . En ese caso, puesto que $F_n(Y) = 1 \ominus F_p(Y)$, la falta se detectará en aquellas entradas Y_k para las que $F_n(Y_k) = 0$ y cambien debido a la falta. La salida que se observe depende de las impedancias relativas de los caminos de conducción en la red de transistores PMOS y en la red de transistores NMOS.

Así pues, el proceso de obtención de patrones de test para las faltas simples de anclaje de transistor es el siguiente:

Procedimiento IV.1:

1. Para cada falta, se selecciona un coeficiente espectral que cambia debido a esa falta.
2. Si la falta es de anclaje abierto:
 - 2.a. Entre las combinaciones de entrada de las que depende el coeficiente seleccionado se toma una, Y_k , de entre aquellas para las que $F(Y_k) = 1$, si se trata de una falta de anclaje en un transistor de la red NMOS, o $F(Y_k) = 0$, si la falta es en un transistor de la red PMOS.
 - 2.b. La secuencia de patrones de test está formada por los valores Y_1 para los que $F(Y_1) = 0$, precedidos por el patrón Y_k seleccionado en 2.a, si la falta de anclaje es en la red NMOS; o los valores Y_1 para los que $F(Y_1) = 1$ precedidos por el patrón Y_k seleccionado en 2.a, si la falta se produce en la red PMOS.
3. Si la falta es de anclaje a corto, se eligen como patrones aquellos Y_k de los que depende el coeficiente espectral seleccionado para los que $F(Y_k) = 1$, si la falta es en la red de transistores NMOS; o aquellos para los que $F(Y_k) = 0$, si la falta es en la red PMOS. ■

Para finalizar este apéndice, se considerará un ejemplo: se pretende detectar una falta de anclaje a abierto en el transistor T6 de la figura IV.1. Esta falta anula los coeficientes $r_n(0110)$, $r_n(0111)$ y $r(1111)$. De entre esos coeficientes se selecciona $r_n(0110)$ porque depende de menor número de combinaciones de entrada. Concretamente, depende de las combinaciones de entradas, $(Y_1, Y_2, Y_3, Y_4) = (0000), (0100), (0010)$ y (0110) . Como $F(0000) = 1$, $F(0100) = 1$, $F(0010) = 1$, $F(0110) = 0$, la secuencia de patrones de test que detectan la falta es $\{(0010), (0110)\}$.

APENDICE V. SINTESIS DEL MODULO EXTRACTOR MEDIANTE UNA RED DE HOPFIELD.

En este apéndice se considera el uso de una red neuronal de Hopfield generalizada [HOP82, 84] en la búsqueda de los coeficientes espectrales que minimizan (6.17).

V.1.Red de Hopfield generalizada.

En la expresión de la probabilidad de enmascaramiento de una de las funciones realizadas por el extractor $p_{A_i} = 1 - \sum_{x, e \neq 0} p(x, e) [\sum'_{j \in S(x, e)} \alpha_j]$ aparece la suma módulo-2 o EXOR (\sum') $\sum'_{j \in S(x, e)} \alpha_j$, que se puede expresar como una suma aritmética si se tiene en cuenta que:

$$\sum'_{i=1}^n A_i = \sum_{i=1}^n (-2)^{i-1} \{ \sum_{r=1}^{w_i} A_{ur(1)} \cdots A_{ur(i)} \} \quad (V.1)$$

donde $w_i = \binom{n}{i}$ y $A_{ur(1)} \cdots A_{ur(i)}$ es la combinación r-ésima de los n elementos A_j ($j = 1, \dots, n$) tomados de i en i.

Para sustituir (V.1) en (6.17) hay que tener en cuenta que el índice j que se mueve dentro de $S(x, e)$ en (6.17) es el que corresponde al índice i en (V.1). Por consiguiente, se supondrá una numeración de los elementos de $S(x, e)$ y se utiliza n para denotar su cardinal. Con ello:

$$p_{Ai} = 1 - \sum_{x, e \neq 0} p(x, e) \left[\sum_{j=1}^n (-2)^{j-1} \left\{ \sum_{s=1}^j w_j \alpha_{\mu_s(1)} \cdots \alpha_{\mu_s(j)} \right\} \right] \quad (V.2)$$

donde $\alpha_{\mu_s(1)} \cdots \alpha_{\mu_s(j)}$ es la combinación s-ésima de los n elementos de $S(x, e)$ tomados de j en j.

En la expresión (V.2) sólo aparecen sumas y productos aritméticos y se puede transformar reagrupando los sumandos para que tome la forma:

$$p_{Ai} = 1 - \sum_{j=1}^{r+1} \sum_{s=1}^j w_j (-2)^{j-1} \phi(\mu_s(1), \dots, \mu_s(j)) \alpha_{\mu_s(1)} \cdots \alpha_{\mu_s(j)}$$

con $w_j = \binom{r+1}{j}$ (V.3)

En (V.3), $\phi(\mu_s(1), \dots, \mu_s(j))$ es la suma de todas las probabilidades $p(x, e)$ con $e \neq 0$ que están multiplicadas en (6.17) por la suma EXOR de los coeficientes espectrales $\alpha_{\mu_s(1)}, \dots, \alpha_{\mu_s(j)}$, conjunto que se notará como $R(\mu_s(1) \dots \mu_s(j))$:

$$\phi(\mu_s(1), \dots, \mu_s(j)) = \sum_{(e, x) \in R(\mu_s(1) \dots \mu_s(j))} p(x, e) \quad (V.4)$$

Se llamará a $\phi(\mu_s(1), \dots, \mu_s(j))$ **potencial de detección de los coeficientes espectrales** $\alpha_{\mu_s(1)}, \dots, \alpha_{\mu_s(j)}$. Para obtener los potenciales de detección asociados a cada grupo de coeficientes, se pueden utilizar las intersecciones de los conjuntos de pares (x, e) obtenidos en la sección 6.2 mediante el algoritmo 6.3.

La ecuación (V.3) se puede hacer corresponder con la ecuación de la energía de una red neuronal generalizada de Hopfield [TSI90], de tantas neuronas como coeficientes tenga el espectro de Tamari de la función g_i sintetizada por el módulo extractor. En esa red se identificarán:

- (a) la energía con la probabilidad de enmascaramiento p_{Ai} ,
- (b) el estado de excitación de la neurona j-ésima con el valor del coeficiente espectral α_j ,
- (c) el umbral de la neurona j-ésima corresponde al potencial de detección del coeficiente α_j , $\phi(j)$.

Además, el peso con el que la entrada $\alpha_{\mu_s(1)} \cdots \alpha_{\mu_s(i)}$ actúa sobre la neurona j-ésima sería $(-2)^i \phi(\mu_s(1) \dots \mu_s(i) j)$.

Obsérvese que las entradas a cada neurona pueden depender, no sólo de las salidas de otras neuronas, sino también del producto de varias de ellas, de ahí el nombre de red de Hopfield generalizada (GNH) [TSI90]. No obstante, si los potenciales de detección para las agrupaciones de más de dos

coeficientes espectrales son despreciables, la red es una red de Hopfield sin más. En la figura V.1 se muestra el esquema de una red de Hopfield.

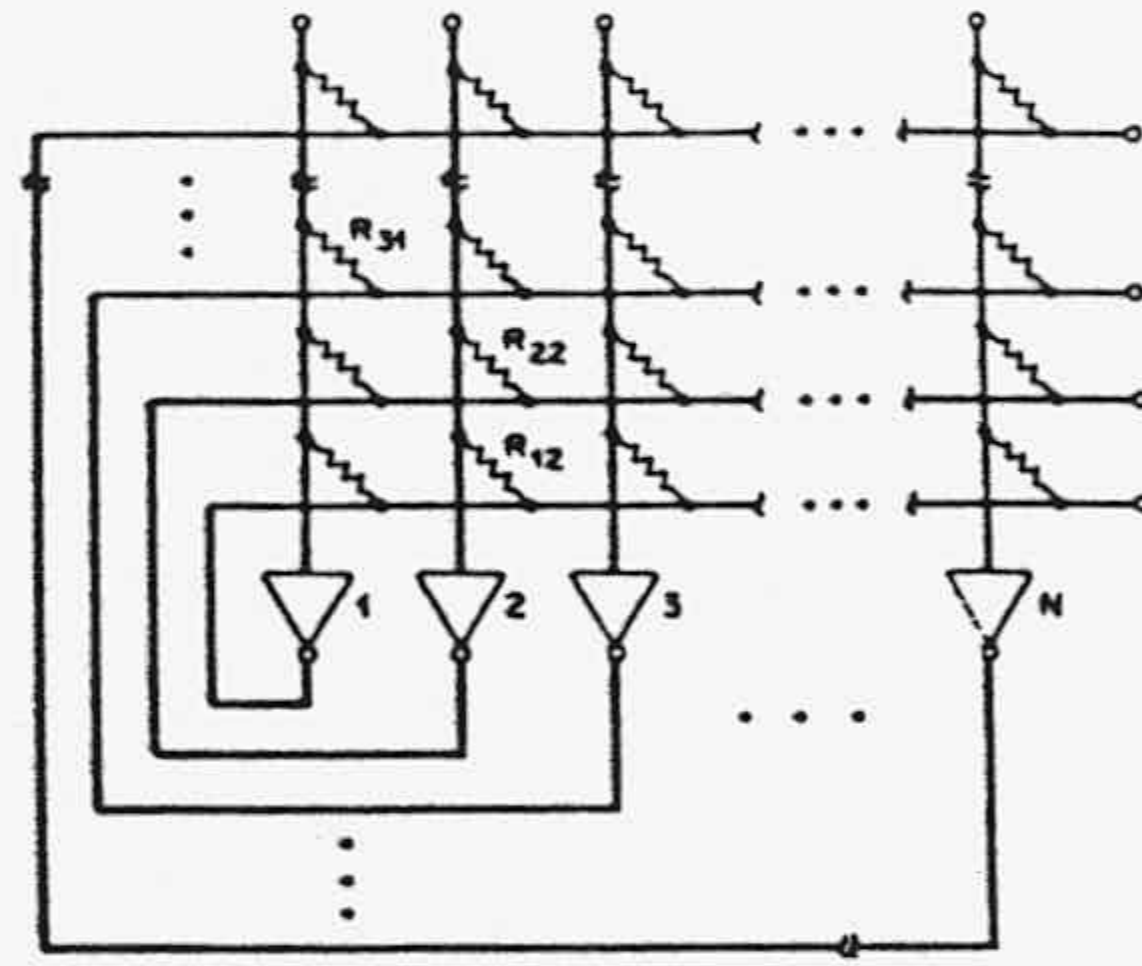


Figura V.1. Red neuronal de Hopfield.

Una red de Hopfield evoluciona hacia su estado de mínima energía a través de cambios en los estados de sus neuronas. Esos cambios son asíncronos (sólo cambia una neurona cada vez) y se demuestra que, efectivamente, gracias a la ley de funcionamiento de cada neurona, la energía de la red disminuye. Veamos qué pasa en la aplicación que se plantea aquí. Para ello se utiliza (V.3) y se supone que se produce un cambio en la neurona j (coeficiente espectral α_j):

$$\delta p_{A1} = -\delta \alpha_j \left(\phi(j) \sum_{k=1}^{r+1} \sum_{s=1}^{w^k} ((-2)^k \phi(j, \mu_s(1), \dots, \mu_s(k)) \alpha_{\mu_s(1)} \dots \alpha_{\mu_s(k)}) \right) \quad (V.5)$$

Para que p_{A1} disminuya, debe ocurrir que $\delta p_{A1} < 0$ y, por consiguiente, que

$$\delta \alpha_j > 0 \text{ si } \phi(j) > \sum_{k=1}^{r+1} \sum_{s=1}^{w^k} ((-2)^k \phi(j, \mu_s(1), \dots, \mu_s(k)) \alpha_{\mu_s(1)} \dots \alpha_{\mu_s(k)})$$

(V.6)

$$\delta \alpha_j < 0 \text{ si } \phi(j) < \sum_{k=1}^{r+1} \sum_{s=1}^{w^k} ((-2)^k \phi(j, \mu_s(1), \dots, \mu_s(k)) \alpha_{\mu_s(1)} \dots \alpha_{\mu_s(k)})$$

La ley de disparo de las neuronas que se muestra en (V.6) es la inversa de la ley de disparo que se utiliza normalmente [HOP82, 84, TSI89]: Aquí las neuronas se disparan cuando la excitación no supera el umbral. Para fijar ideas a continuación se desarrolla un ejemplo.

V.2.Ejemplo.

Dado un circuito de dos salidas, cuyo conjunto de probabilidades de salidas y errores, $p(x,e)$, es:

$$\begin{array}{lll} p(00,01) & p(00,10) & p(00,11) \\ p(01,01) & p(01,10) & p(01,11) \\ p(10,01) & p(10,10) & p(10,11) \\ p(11,01) & p(11,10) & p(11,11) \end{array}$$

se trata de encontrar una función para el módulo de extracción, g_1 , teniéndose:

$$\begin{aligned} g_1(x_1, x_2) &= \alpha_0 \oplus \alpha_1 x_1 \oplus \alpha_2 x_2 \oplus \alpha_3 x_1 x_2 \\ g_1(x_1 \oplus e_1, x_2 \oplus e_2) &= \alpha_0 \oplus \alpha_1 (x_1 \oplus e_1) \oplus \alpha_2 (x_2 \oplus e_2) \oplus \alpha_3 (x_1 \oplus e_1) (x_2 \oplus e_2) \\ g_1(x) \oplus g_1(x \oplus e) &= (\alpha_1 e_1 \oplus \alpha_2 e_2 \oplus \alpha_3 e_1 e_2) \oplus \alpha_3 e_2 x_1 \oplus \alpha_3 e_1 x_2 \end{aligned}$$

La expresión para la probabilidad de enmascaramiento de esta función será:

$$p_{A1} = 1 - \sum_{x, e \neq 0} p(x, e) [(\alpha_1 e_1 \oplus \alpha_2 e_2 \oplus \alpha_3 e_1 e_2) \oplus \alpha_3 e_2 x_1 \oplus \alpha_3 e_1 x_2]$$

y sustituyendo los correspondientes valores de x y e , se tiene:

$$\begin{aligned} p_{A1} = 1 - [& p(00,01)\alpha_1 + p(01,01)\alpha_1 + p(10,01)(\alpha_1 \oplus \alpha_3) + \\ & + p(11,01)(\alpha_1 \oplus \alpha_3) + p(00,01)\alpha_2 + p(01,10)(\alpha_2 \oplus \alpha_3) + \\ & + p(10,01)\alpha_2 + p(11,10)(\alpha_2 \oplus \alpha_3) + p(00,11)(\alpha_1 \oplus \alpha_2 \oplus \alpha_3) + \\ & + p(01,11)(\alpha_1 \oplus \alpha_2) + p(10,11)(\alpha_1 \oplus \alpha_2) + \\ & + p(11,11)(\alpha_1 \oplus \alpha_2 \oplus \alpha_3)] \end{aligned} \quad (V.7)$$

que corresponde a la particularización de la expresión (6.17) para este caso.

Utilizando (V.1), (6.17) pasa a ser:

$$\begin{aligned} p_{A1} = 1 - [& p(00,01)\alpha_1 + p(01,01)\alpha_1 + p(10,01)(\alpha_1 + \alpha_3 - 2\alpha_1\alpha_3) + \\ & + p(11,01)(\alpha_1 + \alpha_3 - 2\alpha_1\alpha_3) + p(00,01)\alpha_2 + \\ & + p(01,10)(\alpha_2 + \alpha_3 - 2\alpha_2\alpha_3) + p(10,01)\alpha_2 + \\ & + p(11,10)(\alpha_2 + \alpha_3 - 2\alpha_2\alpha_3) + \\ & + p(00,11)(\alpha_1 + \alpha_2 + \alpha_3 - 2\alpha_1\alpha_2 - 2\alpha_2\alpha_3 - 2\alpha_1\alpha_3 + 4\alpha_1\alpha_2\alpha_3) + \\ & + p(01,11)(\alpha_1 + \alpha_2 - 2\alpha_1\alpha_2) + p(10,11)(\alpha_1 + \alpha_2 - 2\alpha_1\alpha_2) + \\ & + p(11,11)(\alpha_1 + \alpha_2 + \alpha_3 - 2\alpha_1\alpha_2 - 2\alpha_2\alpha_3 - 2\alpha_1\alpha_3 + 4\alpha_1\alpha_2\alpha_3)] = \\ = 1 - [& (\phi(1)\alpha_1 + \phi(2)\alpha_2 + \phi(3)\alpha_3) - \\ & - 2(\phi(1,2)\alpha_1\alpha_2 + \phi(1,3)\alpha_1\alpha_3 + \phi(2,3)\alpha_2\alpha_3) + \\ & + 4\phi(1,2,3)\alpha_1\alpha_2\alpha_3] \end{aligned} \quad (V.8)$$

que corresponde a la expresión (V.3) para el caso de que haya dos entradas al módulo de extracción. Los potenciales de detección que aparecen en (V.8) se obtienen utilizando las siguientes expresiones:

$$\begin{aligned}
 \phi(1) &= p(00,01)+p(01,01)+p(10,01)+p(11,01)+ \\
 &\quad p(00,11)+p(01,11)+p(10,11)+p(11,11) \\
 \phi(2) &= p(00,10)+p(01,10)+p(10,10)+p(11,10)+ \\
 &\quad p(00,11)+p(01,11)+p(10,11)+p(11,11) \\
 \phi(3) &= p(10,01)+p(11,01)+p(01,10)+p(11,10)+ \\
 &\quad p(00,11)+p(11,11) \\
 \phi(1,2) &= p(00,11)+p(01,11)+p(10,11)+p(11,11)+ \\
 \phi(2,3) &= p(01,10)+p(11,10)+p(00,11)+p(11,11) \\
 \phi(1,3) &= p(10,01)+p(11,10)+p(00,11)+p(11,11)+ \\
 \phi(1,2,3) &= p(00,11)+p(11,11)
 \end{aligned} \tag{V.9}$$

Una vez obtenida la expresión de p_{A1} , los cambios en los estados de las tres neuronas de la red se producen según las expresiones siguientes:

- (1) $\delta\alpha_1 > 0$ si $\phi(1) > [2(\phi(1,2)\alpha_2 + \phi(1,3)\alpha_3) - 4\phi(1,2,3)\alpha_2\alpha_3]$
 $\delta\alpha_1 < 0$ en caso contrario.
- (2) $\delta\alpha_2 > 0$ si $\phi(2) > [2(\phi(1,2)\alpha_1 + \phi(2,3)\alpha_3) - 4\phi(1,2,3)\alpha_1\alpha_3]$
 $\delta\alpha_2 < 0$ en caso contrario.
- (3) $\delta\alpha_3 > 0$ si $\phi(3) > [2(\phi(1,3)\alpha_1 + \phi(2,3)\alpha_2) - 4\phi(1,2,3)\alpha_1\alpha_2]$
 $\delta\alpha_3 < 0$ en caso contrario.

Así, si definimos:

$$\begin{aligned}
 G_1 &= -(\phi(1) - [2(\phi(1,2)\alpha_2 + \phi(1,3)\alpha_3) - 4\phi(1,2,3)\alpha_2\alpha_3]) \\
 G_2 &= -(\phi(2) - [2(\phi(1,2)\alpha_1 + \phi(2,3)\alpha_3) - 4\phi(1,2,3)\alpha_1\alpha_3]) \\
 G_3 &= -(\phi(3) - [2(\phi(1,3)\alpha_1 + \phi(2,3)\alpha_2) - 4\phi(1,2,3)\alpha_1\alpha_2])
 \end{aligned}$$

se producirá un $\delta\alpha_1 > 0$ si $G_1 < 0$ y un $\delta\alpha_1 < 0$ si $G_1 > 0$.

Sean unos valores concretos para las probabilidades de error:

$$\begin{aligned}
 p(00,01) &= p(01,01) = p(10,01) = p(11,01) = 0.1 \\
 p(00,10) &= p(01,10) = p(10,10) = p(11,10) = 0.1 \\
 p(00,11) &= p(01,11) = p(10,11) = p(11,11) = 0.05
 \end{aligned}$$

para los cuales

$$\begin{aligned}
 G_1 &= -(0.6 - 0.4\alpha_2 - 0.6\alpha_3 + 0.4\alpha_2\alpha_3) \\
 G_2 &= -(0.6 - 0.4\alpha_1 - 0.6\alpha_3 + 0.4\alpha_1\alpha_3) \\
 G_3 &= -(0.5 - 0.6\alpha_1 - 0.6\alpha_2 + 0.4\alpha_1\alpha_2)
 \end{aligned}$$

- Se comienza, por ejemplo, con $\alpha = (\alpha_1, \alpha_2, \alpha_3) = (0, 0, 0)$.
- Con esta situación $G_2 = -0.6$ por lo que $\delta\alpha_2 = 1$ y se llega a $\alpha = (0, 1, 0)$.
- Ahora $G_3 = 0.1$ y debe ocurrir que $\delta\alpha_3 < 0$. Como $\alpha_3 = 0$ se mantiene el estado.

- El valor de $G_1 = -0.2$ por lo que $\delta\alpha_1 > 0$ y se tiene $\alpha = (1, 1, 0)$.
- Con esta situación $G_1 < 0$, $G_2 < 0$ y $G_3 > 0$, por lo que los valores de α_1 , α_2 y α_3 no cambian. Esto significa que la función de extracción solución es $g_1 = x_1 \oplus x_2$, como es lógico, puesto que las distribuciones de errores de las que se partió indicaban mayor probabilidad para los patrones de error con un sólo bit erróneo. Con $\alpha_1 = 1$, $\alpha_2 = 1$ y $\alpha_3 = 0$, el valor de la probabilidad de enmascaramiento es $p_{A1} = 0.2$, que corresponde a la probabilidad de que haya más de un bit erróneo en la salida.

Si se desea continuar disminuyendo la probabilidad de enmascaramiento, se hacen cero todas las probabilidades de error que detecta la función g_1 obtenida y se normalizan. En este caso se tendrá:

$$p(00,11) = p(01,11) = p(10,11) = p(11,11) = 0.25$$

y los valores de los G_i son:

$$G_1 = -(1 - 2\alpha_2 - \alpha_3 + 2\alpha_2\alpha_3)$$

$$G_2 = -(1 - 2\alpha_1 - \alpha_3 + 2\alpha_1\alpha_3)$$

$$G_3 = -(0.5 - \alpha_1 - \alpha_2 + 2\alpha_1\alpha_2)$$

Se parte, como antes, de $\alpha = (0, 0, 0)$. Al ser $G_1 < 0$, α_1 puede pasar a ser $\alpha_1 = 1$. Ahora, $G_2 > 0$, $G_3 > 0$ y $G_1 < 0$, puesto que $\alpha_2 = 0$, $\alpha_3 = 0$ y $\alpha_1 = 1$, no se pueden producir cambios. Se ha llegado al estado de equilibrio y

$$g_1 = x_1 \oplus x_2$$

$$g_2 = x_1$$

Como $p_{A2} = 0$, $p_A = p_{A1} p_{A2} = 0$, lo que significa que todo error produce un cambio en las salidas del módulo extractor.

APENDICE VI: LISTADOS DE DESCRIPCION DE RED Y DE SCOAP PARA LA ALU 74181.

VI.1 Descripción de Red de la ALU 74181.

```

* SCOAP /ALU74181/schematic/current
input  N17 N16 N15 +
        N14 N13 N12 +
        N11 N10 N9  +
        N8  N7  N6  +
        N5  N4
output N25 N24 N23 +
        N22 N21 N20 +
        N19 N18
* Net0 = /86.Y
* Net1 = /58.Y
* Net2 = /61.Y
* Net3 = /103.Y
* Net4 = /S0
* Net5 = /S1
* Net6 = /S2
* Net7 = /S3
* Net8 = /B3
* Net9 = /A3
* Net10 = /B2
* Net11 = /A2
* Net12 = /B1
* Net13 = /A1
* Net14 = /B0
* Net15 = /A0
* Net16 = /M
* Net17 = /Cn
* Net18 = /F0
* Net19 = /F1
* Net20 = /AB
* Net21 = /F2

```

* Net22 = /Gc
* Net23 = /Cn4
* Net24 = /Pc
* Net25 = /F3
* Net26 = /0.Y
* Net27 = /6.Y
* Net28 = /19.Y
* Net29 = /28.Y
* Net30 = /52.Y
* Net31 = /53.Y
* Net32 = /59.Y
* Net33 = /60.Y
* Net34 = /72.Y
* Net35 = /84.Y
* Net36 = /90.Y
* Net37 = /97.Y
* Net38 = /2.Y
* Net39 = /29.Y
* Net40 = /55.Y
* Net41 = /57.Y
* Net42 = /62.Y
* Net43 = /71.Y
* Net44 = /1.Y
* Net45 = /20.Y
* Net46 = /23.Y
* Net47 = /56.Y
* Net48 = /51.Y
* Net49 = /63.Y
* Net50 = /98.Y
* Net51 = /3.Y
* Net52 = /5.Y
* Net53 = /17.Y
* Net54 = /18.Y
* Net55 = /54.Y
* Net56 = /64.Y
* Net57 = /91.Y
* Net58 = /77.Y
* Net59 = /83.Y
* Net60 = /89.Y
* Net61 = /73.Y
* Net62 = /85.Y
* Net63 = /78.Y
* Net64 = /93.Y
* Net65 = /99.Y
* Net66 = /102.Y
* Net67 = /82.Y
* Net68 = /74.Y
* Net69 = /80.Y
* Net70 = /87.Y
* Net71 = /94.Y
* Net72 = /100.Y
* Net73 = /4.Y
nand2 N66 N3 N17
* I0=/102
* N66=Y N3=A N17=B
and5 N67 N17 N1 N48 N46 N3
* I1=/82
* N67=Y N17=A N1=B N48=C N46=D N3=E


```

xor2 N18 N72 N66
* I2=/101
* N18=Y N72=A N66=B
xor2 N72 N2 N1
* I3=/100
* N72=Y N2=A N1=B
xor2 N19 N71 N65
* I4=/96
* N19=Y N71=A N65=B
xor2 N21 N70 N64
* I5=/95
* N21=Y N70=A N64=B
xor2 N71 N48 N55
* I6=/94
* N71=Y N48=A N55=B
xor2 N70 N46 N54
* I7=/87
* N70=Y N46=A N54=B
xor2 N25 N69 N0
* I8=/81
* N25=Y N69=A N0=B
xor2 N69 N73 N52
* I9=/80
* N69=Y N73=A N52=B
or2 N23 N58 N63
* I10=/79
* N23=Y N58=A N63=B
nor4 N0 N67 N59 N35 N62
* I11=/86
* N0=Y N67=A N59=B N35=C N62=D
nor4 N22 N52 N43 N34 N61
* I12=/76
* N22=Y N52=A N43=B N34=C N61=D
nand4 N24 N73 N46 N48 N1
* I13=/75
* N24=Y N73=A N46=B N48=C N1=D
nand5 N68 N73 N46 N48 N1 N17
* I14=/74
* N68=Y N73=A N46=B N48=C N1=D N17=E
and4 N20 N25 N21 N19 N18
* I15=/104
* N20=Y N25=A N21=B N19=C N18=D
and4 N60 N17 N1 N48 N3
* I16=/89
* N60=Y N17=A N1=B N48=C N3=D
and4 N59 N48 N46 N2 N3
* I17=/83
* N59=Y N48=A N46=B N2=C N3=D
and4 N61 N73 N46 N48 N2
* I18=/73
* N61=Y N73=A N46=B N48=C N2=D
inv N3 N16
* I19=/103
* N3=Y N16=A
inv N63 N68
* I20=/78
* N63=Y N68=A
inv N58 N22

```

```

* I21=/77
* N58=Y N22=A
inv N33 N14
* I22=/60
* N33=Y N14=A
inv N31 N12
* I23=/53
* N31=Y N12=A
inv N29 N10
* I24=/28
* N29=Y N10=A
inv N27 N8
* I25=/6
* N27=Y N8=A
nor3 N64 N60 N36 N57
* I26=/93
* N64=Y N60=A N36=B N57=C
nor3 N2 N42 N56 N15
* I27=/61
* N2=Y N42=A N56=B N15=C
nor3 N55 N40 N41 N13
* I28=/54
* N55=Y N40=A N41=B N13=C
nor3 N54 N39 N53 N11
* I29=/18
* N54=Y N39=A N53=B N11=C
nor3 N52 N38 N51 N9
* I30=/5
* N52=Y N38=A N51=B N9=C
nor2 N65 N37 N50
* I31=/99
* N65=Y N37=A N50=B
nor2 N1 N32 N49
* I32=/58
* N1=Y N32=A N49=B
nor2 N48 N30 N47
* I33=/51
* N48=Y N30=A N47=B
nor2 N46 N28 N45
* I34=/23
* N46=Y N28=A N45=B
nor2 N73 N26 N44
* I35=/4
* N73=Y N26=A N44=B
and2 N50 N2 N3
* I36=/98
* N50=Y N2=A N3=B
and2 N57 N55 N3
* I37=/91
* N57=Y N55=A N3=B
and2 N62 N54 N3
* I38=/85
* N62=Y N54=A N3=B
and2 N43 N73 N54
* I39=/71
* N43=Y N73=A N54=B
and2 N56 N4 N14
* I40=/64

```

* N56=Y N4=A N14=B
 and2 N42 N33 N5
 * I41=/62
 * N42=Y N33=A N5=B
 and2 N41 N4 N12
 * I42=/57
 * N41=Y N4=A N12=B
 and2 N40 N31 N5
 * I43=/55
 * N40=Y N31=A N5=B
 and2 N53 N4 N10
 * I44=/17
 * N53=Y N4=A N10=B
 and2 N39 N29 N5
 * I45=/29
 * N39=Y N29=A N5=B
 and2 N51 N4 N8
 * I46=/3
 * N51=Y N4=A N8=B
 and2 N38 N27 N5
 * I47=/2
 * N38=Y N27=A N5=B
 and3 N37 N17 N1 N3
 * I48=/97
 * N37=Y N17=A N1=B N3=C
 and3 N36 N48 N2 N3
 * I49=/90
 * N36=Y N48=A N2=B N3=C
 and3 N35 N46 N55 N3
 * I50=/84
 * N35=Y N46=A N55=B N3=C
 and3 N34 N73 N46 N55
 * I51=/72
 * N34=Y N73=A N46=B N55=C
 and3 N49 N15 N6 N33
 * I52=/63
 * N49=Y N15=A N6=B N33=C
 and3 N32 N14 N7 N15
 * I53=/59
 * N32=Y N14=A N7=B N15=C
 and3 N47 N13 N6 N31
 * I54=/56
 * N47=Y N13=A N6=B N31=C
 and3 N30 N12 N7 N13
 * I55=/52
 * N30=Y N12=A N7=B N13=C
 and3 N45 N11 N6 N29
 * I56=/20
 * N45=Y N11=A N6=B N29=C
 and3 N28 N10 N7 N11
 * I57=/19
 * N28=Y N10=A N7=B N11=C
 and3 N44 N9 N6 N27
 * I58=/1
 * N44=Y N9=A N6=B N27=C
 and3 N26 N8 N7 N9
 * I59=/0
 * N26=Y N8=A N7=B N9=C



VI.2 Resultados de SCOAP para el análisis de testeabilidad.

```

***** program scoap *****
version      2.1      11/1/81
*****
*   p r e l i m i n a r y   *
*****

```

input network read from file "netlist"

```

input network node count:  74
floating node count:      0
fanout branch count:     105
total composite node count: 179

```

```

cc0 = zero combinational controllability
cc1 = one combinational controllability
co  = combinational observability
sc0 = zero sequential controllability
sc1 = one sequential controllability
so  = sequential observability

```

node name	cc0	cc1	co	sc0	sc1	so
N0	10	12	9	0	0	0
N1	5	5	6	0	0	0
N1-N24-4 ...	5	5	16	0	0	0
N1-N37-2 ...	5	5	17	0	0	0
N1-N60-2 ...	5	5	25	0	0	0
N1-N67-2 ...	5	5	33	0	0	0
N1-N68-4 ...	5	5	32	0	0	0
N1-N72-2 ...	5	5	6	0	0	0
N10	1	1	19	0	0	0
N10-N28-1 ..	1	1	19	0	0	0
N10-N29-1 ..	1	1	20	0	0	0
N10-N53-2 ..	1	1	21	0	0	0
N11	1	1	19	0	0	0
N11-N28-3 ..	1	1	19	0	0	0
N11-N45-1 ..	1	1	20	0	0	0
N11-N54-3 ..	1	1	20	0	0	0
N12	1	1	16	0	0	0
N12-N30-1 ..	1	1	16	0	0	0
N12-N31-1 ..	1	1	17	0	0	0
N12-N41-2 ..	1	1	19	0	0	0
N13	1	1	16	0	0	0
N13-N30-3 ..	1	1	16	0	0	0
N13-N47-1 ..	1	1	17	0	0	0
N13-N55-3 ..	1	1	18	0	0	0
N14	1	1	12	0	0	0
N14-N32-1 ..	1	1	12	0	0	0

N14-N33-1 ..	1	1	13	0	0	0
N14-N56-2 ..	1	1	15	0	0	0
N15	1	1	12	0	0	0
N15-N2-3 ...	1	1	14	0	0	0
N15-N32-3 ..	1	1	12	0	0	0
N15-N49-1 ..	1	1	13	0	0	0
N16	1	1	12	0	0	0
N17	1	1	12	0	0	0
N17-N37-1 ..	1	1	21	0	0	0
N17-N60-1 ..	1	1	29	0	0	0
N17-N66-2 ..	1	1	12	0	0	0
N17-N67-1 ..	1	1	37	0	0	0
N17-N68-5 ..	1	1	36	0	0	0
N18	11	11	0	0	0	0
N19	15	15	0	0	0	0
N2	2	6	9	0	0	0
N2-N36-2 ...	2	6	23	0	0	0
N2-N50-1 ...	2	6	15	0	0	0
N2-N59-3 ...	2	6	31	0	0	0
N2-N61-4 ...	2	6	25	0	0	0
N2-N72-1 ...	2	6	9	0	0	0
N20	12	64	0	0	0	0
N21	18	18	0	0	0	0
N22	7	12	0	0	0	0
N23	17	9	0	0	0	0
N24	21	6	0	0	0	0
N25	19	19	0	0	0	0
N26	2	4	17	0	0	0
N27	2	2	16	0	0	0
N27-N38-1 ..	2	2	16	0	0	0
N27-N44-3 ..	2	2	20	0	0	0
N28	2	4	16	0	0	0
N29	2	2	19	0	0	0
N29-N39-1 ..	2	2	21	0	0	0
N29-N45-3 ..	2	2	19	0	0	0
N3	2	2	11	0	0	0
N3-N35-3 ...	2	2	30	0	0	0
N3-N36-3 ...	2	2	27	0	0	0
N3-N37-3 ...	2	2	20	0	0	0
N3-N50-2 ...	2	2	19	0	0	0
N3-N57-2 ...	2	2	22	0	0	0
N3-N59-4 ...	2	2	35	0	0	0
N3-N60-4 ...	2	2	28	0	0	0
N3-N62-2 ...	2	2	25	0	0	0
N3-N66-1 ...	2	2	11	0	0	0
N3-N67-5 ...	2	2	36	0	0	0
N30	2	4	13	0	0	0
N31	2	2	16	0	0	0
N31-N40-1 ..	2	2	19	0	0	0

N31-N47-3 ..	2	2	16	0	0	0
N32	2	4	9	0	0	0
N33	2	2	12	0	0	0
N33-N42-1 ..	2	2	15	0	0	0
N33-N49-3 ..	2	2	12	0	0	0
N34	3	17	9	0	0	0
N35	3	14	18	0	0	0
N36	3	14	15	0	0	0
N37	2	9	13	0	0	0
N38	2	4	14	0	0	0
N39	2	4	19	0	0	0
N4	1	1	15	0	0	0
N4-N41-1 ...	1	1	19	0	0	0
N4-N51-1 ...	1	1	16	0	0	0
N4-N53-1 ...	1	1	21	0	0	0
N4-N56-1 ...	1	1	15	0	0	0
N40	2	4	17	0	0	0
N41	2	3	17	0	0	0
N42	2	4	13	0	0	0
N43	3	12	9	0	0	0
N44	2	5	17	0	0	0
N45	2	5	16	0	0	0
N46	5	5	13	0	0	0
N46-N24-2 ..	5	5	16	0	0	0
N46-N34-2 ..	5	5	21	0	0	0
N46-N35-1 ..	5	5	27	0	0	0
N46-N59-2 ..	5	5	32	0	0	0
N46-N61-2 ..	5	5	26	0	0	0
N46-N67-4 ..	5	5	33	0	0	0
N46-N68-2 ..	5	5	32	0	0	0
N46-N70-1 ..	5	5	13	0	0	0
N47	2	5	13	0	0	0
N48	5	5	10	0	0	0
N48-N24-3 ..	5	5	16	0	0	0
N48-N36-1 ..	5	5	24	0	0	0
N48-N59-1 ..	5	5	32	0	0	0
N48-N60-3 ..	5	5	25	0	0	0
N48-N61-3 ..	5	5	26	0	0	0
N48-N67-3 ..	5	5	33	0	0	0
N48-N68-3 ..	5	5	32	0	0	0
N48-N71-1 ..	5	5	10	0	0	0
N49	2	5	9	0	0	0
N5	1	1	16	0	0	0
N5-N38-2 ...	1	1	17	0	0	0
N5-N39-2 ...	1	1	22	0	0	0
N5-N40-2 ...	1	1	20	0	0	0
N5-N42-2 ...	1	1	16	0	0	0
N50	3	9	12	0	0	0

N51	2	3	14	0	0	0
N52	2	6	10	0	0	0
N52-N22-1 ..	2	6	10	0	0	0
N52-N69-2 ..	2	6	17	0	0	0
N53	2	3	19	0	0	0
N54	2	6	15	0	0	0
N54-N43-2 ..	2	6	15	0	0	0
N54-N62-1 ..	2	6	21	0	0	0
N54-N70-2 ..	2	6	16	0	0	0
N55	2	6	13	0	0	0
N55-N34-3 ..	2	6	20	0	0	0
N55-N35-2 ..	2	6	26	0	0	0
N55-N57-1 ..	2	6	18	0	0	0
N55-N71-2 ..	2	6	13	0	0	0
N56	2	3	13	0	0	0
N57	3	9	15	0	0	0
N58	13	8	4	0	0	0
N59	3	19	18	0	0	0
N6	1	1	13	0	0	0
N6-N44-2 ...	1	1	21	0	0	0
N6-N45-2 ...	1	1	20	0	0	0
N6-N47-2 ...	1	1	17	0	0	0
N6-N49-2 ...	1	1	13	0	0	0
N60	2	14	16	0	0	0
N61	3	22	9	0	0	0
N62	3	9	18	0	0	0
N63	3	23	14	0	0	0
N64	10	9	9	0	0	0
N65	10	6	9	0	0	0
N66	4	2	9	0	0	0
N67	2	19	19	0	0	0
N68	22	2	15	0	0	0
N69	8	8	11	0	0	0
N7	1	1	12	0	0	0
N7-N26-2 ...	1	1	20	0	0	0
N7-N28-2 ...	1	1	19	0	0	0
N7-N30-2 ...	1	1	16	0	0	0
N7-N32-2 ...	1	1	12	0	0	0
N70	8	8	10	0	0	0
N71	8	8	7	0	0	0
N72	8	8	3	0	0	0
N73	5	5	14	0	0	0
N73-N24-1 ..	5	5	16	0	0	0
N73-N34-1 ..	5	5	21	0	0	0
N73-N43-1 ..	5	5	16	0	0	0
N73-N61-1 ..	5	5	26	0	0	0
N73-N68-1 ..	5	5	32	0	0	0
N73-N69-1 ..	5	5	14	0	0	0

N8	1	1	16	0	0	0
N8-N26-1 ...	1	1	20	0	0	0
N8-N27-1 ...	1	1	17	0	0	0
N8-N51-2 ...	1	1	16	0	0	0
N9	1	1	15	0	0	0
N9-N26-3 ...	1	1	20	0	0	0
N9-N44-1 ...	1	1	21	0	0	0
N9-N52-3 ...	1	1	15	0	0	0

BIBLIOGRAFIA

- [ABA83a] Abadir, M.S.; Reghbaty, H.K.: "Functional Testing of Semiconductor Random Access Memories". ACM Computing Surveys, Vol. 15, pp. 175-198, September 1983.
- [ABA83b] Abadir, M.S.; Reghbaty, H.K.: "LSI Testing Techniques". IEEE Micro, pp.34-51. February, 1983.
- [ABR80] Abramovici, M.; Breuer, M.A.: "Multiple Fault Diagnosis in Combinational Circuits Based on an Effect-Cause Analysis". IEEE Transactions on Computers, Vol. C-29, No. 6, pp. 451-459, June 1980.
- [ABR84] Abramovici, M.; Menon, P.R.; Miller, D.T.: "Critical Path Tracing: An Alternative to Fault Simulation". IEEE Design & Test of Comp., Vol.1, pp.83-93. February, 1984.
- [ABR86a] Abramovici, M.; Menon, P.R.; Miller, D.T.: "Checkpoint Faults are not Sufficient Target Faults for Test Generation". IEEE Transactions on Computers, Vol. C-35, No. 8, pp. 769-771, August 1986.
- [ABR86b] Abraham, J.A.; Fuchs, W.K.: "Fault and Error Models for VLSI". Proceedings of the IEEE, Vol. 74, No. 5, pp. 639-654, May 1986.
- [AGR75] Agrawal, P.; Agrawal, V.D.: "Probabilistic Analysis of Random Test Generation Method for Irredundant Combinational Logic Networks". IEEE Trans. on Comp., vol.C-24, No.7. July, 1975.
- [AGR81] Agrawal, V.D.: "Sampling Techniques for Determining Fault Coverage in LSI Circuits". Jour. Digital Systems, Vol. V, pp. 189-202, 1981.
- [AGR88] Agrawal, V.D.; Seth, S.C.: "Test Generation for VLSI Chips". IEEE Computer Society Tutorial, No.786. 1988.
- [AHR84] Ahrens, D.P.; Bednarczy, P.J.; Denburg, D.L.; Robertson, R.M.: "TPG2 - An Automatic Test Program Generation for Custom ICs". Proc. Int. Test Conf., pp.762-767. October, 1984.
- [ALA87] Al-Arian, S.A.; Agrawal, D.P.: "Physical Failures and Fault Models of CMOS Circuits". IEEE Transactions on Circuits and Systems, Vol. CAS-34, No. 3, pp. 269-279, March 1987.
- [AND80] Ando, H.: "Testing VLSI with Random Access Scan". Proceedings of COMPCON S'80, pp. 50-52, 1980.
- [ANT87] Antreich, K.J., Schulz, M.H.: "Accelerated Fault Simulation and Fault Grading in Combinational Circuits". IEEE Transactions on Computer-Aided Design, Vol. CAD-6, No. 5, pp. 704-711, September 1987.

- [ARM66] Armstrong, D.B.: "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets". IEEE Transactions on Electronics Computers, Vol. EC-15, No. 1, pp. 66-73, February 1966.
- [ARM72] Armstrong, D.B.: "A Deductive Method for Simulating Faults in Logic Circuits". IEEE Transactions on Computers, Vol. C-21, No. 5, pp. 464-471, May 1972.
- [AVI82] Avizienis, A.: "The Four-Universal Information System Model for the Study of Fault-Tolerance". Proceedings of the Fault-Tolerant Computing Symposium, pp. 6-13, 1982.
- [BAR81] Bardell, P.H.; McAnney, W.H.: "A View from the Trenches: Production Testing of a Family of VLSI Multichip Modules". Proceedings of the Fault-Tolerant Computing Symposium, pp. 281-283, 1981.
- [BAR81] Barzilai, Z.; Savir, J.; Markowsky; Smith, M.G.: "The Weighted Syndrome Sum Approach to VLSI Testing". IEEE Trans. on Comp., vol.C-30, No.12, pp.996-1000. December, 1981.
- [BAR82] Bardell, P.H.; McAnney, W.H.: "Self-Testing of Multichip Logic Modules". Proceedings of the International Test Conference, pp. 200-204, 1982.
- [BAR90] Barton, J.H.; Czeck, E.W.; Segall, Z.Z.; Siewiorek, D.P.: "Fault Injection Experiment Using FIAT". IEEE Trans. on Comp., vol.39, No.4, pp.575-582. April, 1990.
- [BEC89] Bechta, J.; Trivedi, K.S.: "Coverage Modeling for Dependability Analysis of Fault-Tolerant Systems". IEEE Trans. on Comp., vol.38, No.6, pp.775-787. June, 1989.
- [BEN75] Benowitz, N.; Calhoun, D.F.; Alderson, G.E.; Bauer, J.E.; Joeckel, C.T.: "An Advanced Fault Isolation System for Digital Logic". IEEE Transactions on Computers, pp. 489-497, May 1975.
- [BEN84] Bennets, R.G.: "Design of Testable Logic Circuits". Addison-Wesley Pub. Company. 1984.
- [BER73] Berger, I.; Kohavi, K.: "Fault Detection in Fanout-Free Combinational Networks". IEEE Transactions on Computers, Vol. C-22, No. 10, pp. 908-914, October 1973.
- [BER87] Berg, M.; Koren, I.: "On Switching Policies for Modular Redundancy Fault-Tolerant Computing Systems". IEEE Transactions on Computers, Vol. C-36, No. 9, pp. 1052-1062, September 1987.
- [BES78] Besslich, P.W.: "On the Walsh-Hadamard Transform and Prime Implicant Extraction". IEEE Trans on Elect. Comp., vol EMC-20, pp.516-519. 1978.
- [BES84] Beste, W.E. Den: "Using a Software Emulator to Generate and Edit VLSI Test Patterns". Electronics Tests, pp. 42-52, March 1984.
- [BES85] Besslich, P.W.: "Spectral Processing of Switching Functions Using Signal-Flow Transformations", in "Spectral Techniques and Fault Detection", M.G. Karpovsky ed., pp.91-142, 1985.
- [BEY83] Beyers, J.W.; Dohse, L.J.; Fucetola, J.P.; Kolesar, M.L.; Lob, C.G.; Maitland, D.S.; Malhotra, A.K.; Seccombe, S.D.; Wheeler, J.K.; Zeller, E.R.: "A 32-Bit VLSI System". Proceedings of COMPCON S'83, pp. 218-221.
- [BLA88] Blaum, M.: "Systematic Unidirectional Burst Detecting Codes". IEEE Transactions on Computers, Vol. 37, No. 4, pp. 453-457, April 1988.
- [BLO90] Blough, D.M.; Masson, G.M.: "Performance Analysis of a Generalized Concurrent Error Detection Procedure". IEEE Trans. on Comp., Vol.39, No.1, pp.47-62. January, 1990.
- [BOR87] Borden, J.M.; Vinck, A.J.: "On Coding for "Stuck-At" Defects". IEEE Transactions on Information Theory, Vol. IT-33, No. 5, pp. 729-735, September 1987.
- [BOS82a] Bose, P.; Abraham, J.A.: "Test Generation for Programmable Logic Arrays". Proc. 19th Design Automation Conf., pp. 574-580, June 1982.
- [BOS82b] Bose, A.K.; Kozak, P.; Lo, C.-Y.; Nham, H.M.; Pacas-Skewes, E.; Wu, K.: "A Fault Simulator for MOS LSI Circuits". Proceedings of the Design Automation Conference, pp. 400-409, 1982.
- [BOS82c] Bose, B.; Rao, T.R.N.: "Theory of Unidirectional Error Correcting/Detecting Codes". IEEE Transactions on Computers, Vol. C-31, No. 6, pp. 521-530, June 1982.

- [BOS84d] Bose, B.; Lin, D. J.: "PLA Implementation of k-out-of-n Code TSC Checker". IEEE Transactions on Computers, Vol. C-33, No. 6, pp. 583-588, June 1984.
- [BOT77] Bottorff, P.S.; France, R.E.; Garges, N.H.; Orosz, E.J.: "Test Generation for Large Logic Networks". Proceedings of the Design Automation Conference, pp. 479-485, 1977.
- [BRE76] Breuer, M.A.; Friedman, A.D.: "Diagnosis & Reliable Design of Digital Systems". Computer Science Press, 1976.
- [BRG84] Brglez, F.: "On Testability Analysis of Combinational Networks". Proc. Int. Symp. Circ. and Syst., pp. 221-225, May 1984.
- [BRG85] Brglez, F.; Pownall, P.; Hum, R.: "Accelerated ATPG and fault grading via testability analysis". Proc. IEEE International Symposium on Circuits and Systems, June 5-6, 1985, Kyoto, Japan.
- [BRG89a] Brglez, F.; Bryan, D.; Calhoun, J.; Kedem, G.; Lisanke, R.: "Automated Synthesis for Testability". IEEE Transactions on Industrial Electronics, Vol. 36, No. 2, pp. 263-277, May 1989.
- [BRG89b] Brglez, F.; Bryan, D.; Kozminski, K.: "Combinational Profiles of Sequential Benchmark Circuits". 1989 International Symposium on Circuits and Systems, Portland, Oregon, May 9-11, 1989.
- [BUR85] Burgess, N.; Damper, R.I.; Shaw, S.J.; Wilkins, D.R.J.: "Faults and fault effects in NMOS circuits - impact on design for testability". IEE Proc., vol.132, Pt.G, No.3. June, 1985.
- [BUR88] Burgess, N.; Damper, R.I.; Totton, K.A.; Shaw, S.J.: "Physical faults in MOS circuits and their coverage by different fault models". IEE Proceedings, Vol. 135, Pt. E, No. 1, pp. 1-9, January 1988.
- [CHA74] Chang, H.V.-P.; Chappell, S.G.; Elmendorf, C.H.; Schmidt, L.D.: "Comparison of Parallel and Deductive Fault Simulation Methods". IEEE Trans. on Comp., vol.C-23, No.11, pp.1132-1138. November, 1974.
- [CHA78] Cha, C.W.: "A Testing Strategy for PLAs". Proceedings of the Design Automation Conference, pp. 326-334, 1978.
- [CHA88] Chamberlain, R.D.; Edelman, M.N.; Franklin, M.A.; Witte, E.E.: "Simulated Annealing on a Multiprocessor". IEEE Int. Conf.Comp.Des., pp.540-544, 1988.
- [CHA89] Chang, T.-Y.; Wey, C.-L.: "Design of Fault Diagnosable and Repairable PLA's". IEEE Journal of Solid-State Circuits, Vol. 24, No. 5, pp. 1451-1454, October 1989.
- [CHE88] Chen, C.L.: "Exhaustive Test Pattern Generation Using Cyclic Codes". IEEE Transactions on Computers, Vol. 37, No. 2, pp. 225-227, February 1988.
- [CHI87] Chin, C.C.; McCluskey, E.J.: "Test Length for Pseudorandom Testing". IEEE Transactions on Computers, Vol. C-36, No. 2, pp. 252-256, February 1987.
- [CRO80] Crouzet, Y.; Landrault, C.: "Design of Self-Checking MOS-LSI Circuits: Application to a Four-Bit Microprocessor". IEEE Transactions on Computers, Vol. C-29, No. 6, pp. 532-537, June 1980.
- [DAN85] Daniels, R.G.; Bruce, W.C.: "Built-In Self-Test Trends in Motorola Microprocessors". IEEE Des.& Test, pp.64-71. April, 1985.
- [DAR89b] Darmala, T.; Karpovsky, M.G.: "Detection of stuck-at and bridging faults in Reed-Muller Canonical (RMC) networks". IEE Proc., Vol136, Pt.E, No.5. September, 1989.
- [DAR89] Darmala, T.R.; Karpovski, M.: "Fault Detection in Combinational Networks by Reed-Muller Transforms". IEEE Trans. on Comp., vol.38, No.6, pp.788-797. June, 1989.
- [DAS80] Dasgupta, S.; Walther, R.G.; Williams, T.W.; Eichelberger, E.B.: "An Enhancement to LSSD and Some Applications of LSSD in Reliability, Availability, and Serviceability". Proceedings of the Fault-Tolerant Computing Symposium, pp. 32-34, 1980.
- [DAS82] Dasgupta, S.; Goel, P.; Walther, R.G.; Williams, T.W.: "A Variation of LSSD and Its Implications on Design and Test Pattern Generation in VLSI". Proceedings of the International Test Conference, pp. 63-66, 1982.
- [DAV80] David, R.: "Testing by Feedback Shift Register". IEEE Transactions on Computers, Vol. C-29, No. 7, pp. 668-673, July 1980.

- [DAV90] David, R.: "Comments on 'Signature Analysis for Multiple Output Circuits'". IEEE Trans. on Comp., vol39, No.2, pp.287-288. February, 1990.
- [DES77] Deschamps, J.P.;Thayse, A.: "The Module Structure of Discrete Functions". The 7th ISMVL, pp.14-19. Mayo 1977.
- [EAS77] Eastbrook, J.T.; Bennetts, R.G.: "Failure mechanisms in logic circuits and their related fault effects". Proc. IEEE Conf. on New Develop. en Autom. Test., pp.44-47. November, 1977.
- [EDW75] Edwards, C.R.: "The Application of the Rademacher-Walsh Transform to Boolean Functioin Classification and Threshold Logic Synthesis". IEEE Transactions on Computers, pp.48-62, January 1975.
- [EIC77] Eichelberger, E.B.; Williams, T.W.: "A Logic Design Structure for LSI Testability". Proceedings of the 14th Design Automation Conference, pp. 462-468, 1977.
- [ELL90] Elliot, I.D.; Sayers, I.L.: "Implementation of 32-bit RISC processor incorporating hardware concurrent error detection and correction". IEE proc., vol.137, Pt.E, No.1. January, 1990.
- [ERI78] Eris, E.: "Relationships between Rademacher-Walsh spectra of Boolean functions". Computers and Digital Techniques, Vol. 1, No. 2, pp. 45-48, May 1978.
- [ERI84] Eris, E.; Muzio, J.C.: "Syndrome and Autocorrelation-Testable Internally Unate Combinational Networks". Electronic Letters, vol.20, No.6, pp.264-266. 1984.
- [ERI86] Eris, E.; Muzio, J.C.: "Spectral testing of circuit realisations based on linearisations". IEE Proceedings, Vol. 133, Pte. E, No. 2, pp. 73-78, March 1986.
- [EVE67] Eve, S.; Kohavi, I.; Paz, A.: "On minimal modulo-2 sums of products for switching functions". IEEE Trans. Elec. Comp., vol.EC-16, pp.671-674. October, 1967.
- [EVE67] Even, S.; Kohavi, I.; Paz, A.: "On minimal modulo-2 sums of products for switching functions". IEEE Trans. Electron. Comp., Vol. EC-16, pp.671-674. October, 1967.
- [FAN85] Fantini, F.; Morandi, C.: "Failure modes and mechanisms for VLSI ICs-a review". IEE Proceedings, Vol. 132, Pt. G, No. 3, pp. 74-80, June 1985.
- [FAS80] Fasang, P.P.: "BIDCO Built-In Digital Circuit Observer". The Proceedings of the International Test Conference, pp. 261-266, 1980.
- [FUC87] Fuchs, W.K.; Chen, C.-Y. R.; Abraham, J.A.: "Concurrent Error Detection in Highly Structured Logic Arrays". IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 4, pp. 583-593, August 1987.
- [FUJ80] Fujiwara, H; Kinoshita, K.; Ozaki, H.: "Universal Test Sets for Programmable Logic Arrays". Proceedings of the International Fault-Tolerant Computing Symposium, pp. 137-142, 1980.
- [FUJ81] Fujiwara, H., Kinoshita, K.: "A Design of Programmable Logic Arrays with Universal Tests". IEEE Transactions on Computers, Vol. C-30, No. 11, pp. 823-828, November, 1981.
- [FUJ82] Fujiwara, H.; Toida, S.: "The Complexity of Fault Problems for Combinational Logic Circuits". IEEE Trans. on Comp., vol. C-31, No.6. June, 1982.
- [FUJ83] Fujiwara, H., Shimono, T.: "On the Acceleration of Test Generation Algorithms". IEEE Transactions on Computers, Vol. C-32, No. 12, pp. 1137-1144, December 1983.
- [FUJ84] Fujiwara, E.; Mutoh, N.; Matsuoka, K.: "A Self-Testing Group-Parity Prediction Checker and Its Use for Built-In Testing". IEEE Transactions on Computers, Vol. C-33, No. 6, pp. 578-583, June 1984.
- [FUJ87] Fujiwara, E.; Matsuoka, K.: "A Self-Checking Generalized Prediction Checker and Its Use for Built-In Testing". IEEE Transactions on Computers, Vol. C-36, No. 1, pp. 86-93, January 1987.
- [FUJ90] Fujiwara, H.: "Computational Complexity of Controlability/Observability Problems for Combinational Circuits". IEEE Trans. on Comp., Vol.39, No.6. June, 1990.
- [FUN78] Funatsu, S.; Wakatsuki, N.; Yamada, A.: "Designing Digital Circuits with Easily Testable Consideration". Proceedings of the International Test Conference, pp. 98-102, 1978.
- [FUN85] Funatsu, S.; Kawai, M.: "An Automatic Test-Generation System for Large Digital Circuits". IEEE Design & Test, pp.54-60, October 1985.

- [GAL80] Galiay, J.; Crouzet, Y.; Vergniault, M.: "Physical Versus Logical Fault Models MOS LSI Circuits: Impact on their Testability". IEEE Trans. on Comp., vol.C-29, No.6, pp.527-532. June, 1980.
- [GEL87] Gelsinger, P.P.: "Design and Test of the 80386". IEEE Design & Test, pp.42-50. June, 1987.
- [GHA82] Ghate, P.B.: "Electromigration-induced failures in VLSI applications". Proc. Reliability Physics Symp., pp.292-299. March, 1982.
- [GOE80] Goel, P.: "Test Generation Costs Analysis and Projections". Proceedings of the Design Automation Conference, pp. 77-84, 1980.
- [GOE81] Goel, P.: "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits". IEEE Transactions on Computers, Vol. C-30, pp. 215-222, March 1981.
- [GOE82] Goel, N.S.; Karpovsky, M.G.: "Functional testing of computer hardware and data-transmission channels based on minimising the magnitude of undetected errors". IEEE Proceedings, Vol. 129, Pt. E, No. 5, pp. 169-180, September 1982.
- [GOL79] Goldstein, L.H.: "Controllability/Observability Analysis of Digital Circuits". IEEE Transactions on Circ. and Syst., Vol. CAS-26, pp. 685-693, September 1979.
- [GOL80] Gold, L.H.; Thigpen, E.L.: "SCOAP: Sandia Controlability/Obsevability Analysis Program". Proc. 17th Des.Autom. Conf, pp.190-196. June, 1980.
- [GOL88] Golan,P., Novak,O., Hlavicka,J.: "Pseudoexhaustive Test Pattern Generator with Enhanced Fault Coverage". IEEE Transactions on Computers, Vol. 37, No. 4, pp. 496-499, April 1988.
- [GRA82] Grassl, G.; Pflaiderer, H.J.: "A Self-Testing PLA". Proceedings of the International Solid-State Circuits Conference, pp. 60-61, 1982.
- [GRE86] Green, D.H.: "Modern Logic Design". Addison-Wesley, 1986.
- [GUI87] Guima,T.A., Tapia,M.A.: "Differential Calculus for Fault Detection in Multivalued Logic Networks". Proc. IEEE 17th ISMVL, pp. 99-108, 1987.
- [GUP88] Gupta, S.K.; Pradhan, D.J.: "A New Framework for Designing & Analyzing BIST Techniques: Computation of Exact Aliasing Probability. 1988 International Test Conference, pp. 329-342, 1988.
- [HA88] Ha,D.S., Reddy,S.M.: "On the Design of Pseudoexhaustive Testable PLA's". IEEE Transactions on Computers, Vol. 37, No. 4, pp. 468-471, April 1988.
- [HAS84] Hassan, S.Z.: "Signature Testing of Sequential Machines". IEEE Transactions on Computers, Vol. C-33, No. 8, pp. 762-764, August 1984.
- [HAY78] Hayes,J.P.: "Generation of Optimal Transition Count Tests". IEEE Transactions on Computers, Vol. C-27, No. 1, pp. 36-41, January 1978.
- [HAY87] Hayes,J.P.: "An Introduction to Switch-Level Modeling". IEEE Design & Test, pp. 18-25, August 1987.
- [HOP82] Hopfield, J.J.: "Neural Networks and physical systems with emergent collective computational abilities". Proc. of the National Academy of Sciences, USA 79, pp.2554-2558. 1982.
- [HOP84] Hopfield, J.J.: "Neurons with graded response have collective computational properties like those of two-state neurons". Pro. of the National Academy of Sciences USA 81, pp.3088-3092. May, 1984.
- [HSI84] Hsiao, T.-C.; Seth, S.C.: "An Analysis of the Use of Rademacher-Walsh Spectrum in Compact Testing". IEEE Transactions on Computers, Vol. C-33, No. 10, pp. 934-937, October 1984.
- [HUA84] Hua, K.A.; Jou, J.-Y.; Abraham, J.A.: Built-in tests for VLSI finite-state machines". Proc. IEEE 14th FTCS Conf., pp.292-297. 1984.
- [HUG84] Hughes, J.L.A.; McCluskey, E.J.; Lu, D.J.: "Design of Totally Self-Checking Comparators with and Arbitraty Number of Inputs". IEEE Transactions on Computers, Vol. C-33, No. 6, pp. 546-550, June 1984.
- [HUG86] Hughes, J.L.A.; McCluskey, E.J.: "Multiple Stuck-at Fault Coverage of Single Stuck-at Fault Test Sets". Proc.Int.Test Conf., Washington, DC., pp.368-374, September, 1987.
- [HUG88] Hughes,J.L.A.: "Multiple Fault Detection Using Single Fault Test Sets". IEEE Transactions on Computer-Aided Design, Vol. 7, No. 1, pp. 100-108, January 1988.

- [HUR77] Hurst, S.L.: "Detection of Symmetries in Combinational Functions by Spectral Means". IEEE J. on Elect. Circ. and Syst., Vol.1, pp.173-180, 1977.
- [HUR78] Hurst, S.L.: "The logical Processing of Digital Signals". Crane-Russak, N.Y. and Edward Arnold, London, 1978.
- [HUR85] Hurst, S.L.; Miller, D.M.; Muzio, J.C.: "Spectral Techniques in Digital Logic". Academic Press, 1985.
- [HUR86] Hurst, S.L.: "The interrelationships between fault signatures based upon counting techniques". Proceedings of Technical Workshop: New Directions for IC Testing". pp. 6-1-6-21. March 18-20, 1986, Victoria, B.C. Canada.
- [HWA85] Hwang, K.; Briggs, T.A.: "Computer Architecture and Parallel Processing". McGraw Hill, 1985.
- [IBA75] Ibarra, O.H.; Sahni, S.K.: "Polynomially Complete Fault Detection Problems". IEEE Transactions on Computers, Vol. C-24, No. 3, pp. 242-249, March 1975.
- [IWA88] Iwasaki, K.: "Analysis and Proposal of Signature Circuits for LSI Testing". IEEE Transactions on Computer-Aided Design, Vol. 7, No. 1, pp. 84-90, January 1988.
- [JAC87] Jacob, J.; Biswas, N.N.: "GTBD Faults and Lower Bounds on Multiple Fault Coverage of Single Fault Test Sets". Proc.Int. Test Conf., Washington, DC, pp.849-855, September, 1987.
- [JAI85a] Jain, S.K.; Agrawal, V.D.: "Statistical Fault Analysis". IEEE Design & Test of Computers, Vol. 2, pp. 38-44, February 1985.
- [JAI85b] Jain, S.K., Agrawal, V.D.: "Modeling and Test Generation Algorithms for MOS Circuits". IEEE Transactions on Computers, Vol. C-34, No. 5, pp. 426-433, May 1985.
- [JAI86] Jain, S.K., Stroud, Ch.E.: "Built-in Self Testing of Embedded Memories". IEEE Design & Test, pp.27-37, October 1986.
- [JAY88] Jayarama, R.; Darema, F.: "Error Tolerance in Parallel Simulated Annealing Techniques". IEEE Int. Conf.Comp.Des., pp.545-548, 1988.
- [JHA86] Jha, N.K.: "Detecting Multiple Faults in CMOS Circuits". Proc.Int. Test Conf., Washington, DC, pp.514-519, September, 1986.
- [JHA88a] Jha, N.K.: "Multiple Stuck-Open Fault Detection in CMOS Logic Circuits". IEEE Transactions on Computers, Vol. 37, No. 4, pp. 426-432, April 1988.
- [JHA88b] Jha, N.K.: "Testing for Multiple Faults in Domino-CMOS Logic Circuits". IEEE Transactions on Computer-Aided Design, Vol. 7, No. 1, pp. 109-116, January 1988.
- [JHA89] Jha, N.K.: "A Totally Self-Checking Checker for Borden's Code". IEEE Transactions on Computer-Aided Design, Vol. 8, No. 7, pp. 731-736, July 1989.
- [KAR76] Karpovsky, M.G.: "Finite Orthogonal Series in the Design of Digital Systems". Wiley, N.Y., 1976.
- [KAR77a] Karpovsky, M.G.: "Error Detection in Digital Devices and Computer Programs with the Aid of Linear Recurrent Equations Over Finite Commutative Groups". IEEE Transactions on Computers, Vol. C-26, No. 3, pp. 208-218, March 1977.
- [KAR77b] Karpovsky, M.G.; Trachtenberg, E.A.: "Some Optimization Problems for Convolution Systems over Finite Groups". Information and Control, Vol. 34, No. 3, pp. 227-247, July 1977.
- [KAR79] Karpovsky, M.G.: "On Weight Distribution for Binary Linear Codes". IEEE Trans. on Inf. Theory, vol.IT-25, pp.105-109, 1979.
- [KAR79] Karpovsky, M.G., Trachtenberg, E.A.: "Fourier Transform over Finite Groups for Error Detection and Error Correction in Computation Channels", Information and Control, Vol. 40, No. 3, pp. 335-356, March 1979.
- [KAR80] Karpovsky, M.G.; Su, S.Y.H.: "Detection and Location of Input and Feedback Bridging Faults Among Input and Output Lines". IEEE Trans. on Comp., vol.C-29, No.6, pp.523-527. June, 1980.
- [KAR81] Karpovsky, M.G.: "Spectral Methods for Decomposition, Design and Testing of Multiple-Valued Logical Networks". Proc. IEEE 11th ISMVL; pp. 1-9, 1981.
- [KAR81] Karpovsky, M.G.: "Weight Distribution of Translates, Covering Radius and perfect Codes correcting Errors of the Given Weights". IEEE Trans on Inf. Theory, pp.962-972. July, 1981.

- [KAR83] Karpovsky, M.: "Universal Tests for Detection of Input/Output Stuck-At and Bridging Faults". IEEE Transactions on Computers, Vol. C-32, No. 12, pp. 1194-1200, December 1983.
- [KAR85] Karpovsky, M.G.: "Spectral Techniques and Fault Detection". Notes and Reports in Computer Science and Applied Mathematics, Academic Press, 1985.
- [KAR90] Karpovsky, M.G.; Nagvajara, P.: "Optimal Robust Compression of Test Responses". IEEE Trans. on Comp., pp.138-140. January, 1990.
- [KHA82] Khakbaz, J.; McCluskey, E.J.: "Concurrent Error Detection and Testing for Large PLA's". IEEE Journal of Solid-State Circuits, Vol. SC-17, No. 2, pp. 386-394, April 1982.
- [KHA82] Khakbaz, J.; McCluskey, E.J.: "Concurrent Error Detection and Testing for Large PLAs". IEEE Journ. of Solid-State Circ., Vol. SC-17, No.2. April, 1982.
- [KHA84a] Khakbaz, J.: "A Testable PLA Design with Low Overhead and High Fault Coverage". IEEE Transactions on Computers, Vol. C-33, No. 8, pp. 743-753, August 1984.
- [KHA84b] Khakbaz, J.; McCluskey, E.J.: "Self-Testing Embedded Parity Checkers". IEEE Transactions on Computers, Vol. C-33, No. 8, pp. 753-756, August 1984.
- [KIR83] Kirkpatrick, S.; Gelatt, J.; Vecchi, M.P.: "Optimization by Simulated Annealing". Science, 220, pp.671-680. 1983.
- [KNU85] Knuth, D.E.: "El arte de programar ordenadores. Algoritmos fundamentales. Vol.I". Ed. Reverté, 1985.
- [KON79] Könemann, B.; Mucha, J.; Zwiehoff, G.: "Built-In Logic Block Observation Techniques". Proceedings of the International Test Conference, pp. 37-41, 1979.
- [KOR86] Koren, I., Pradhan, D.K.: "Yield and Performance Enhancement Through Redundancy in VLSI and WSI Multiprocessor Systems". Proceedings of the IEEE, Vol. 74, No. 5, pp. 699-711, May 1986.
- [KUB84] Kuban, J.R., Bruce, W.C.: "Self-Testing the Motorola MC6804P2". IEEE Design & Test, pp. 33-41, May 1984.
- [KUN78] Kung, H.T.; Leiserson, C.E.: "Systolic Arrays (for VLSI)", Spare Matrix Proc., Duff et al. eds., Society of Indust. and Appl. Math., pp.245-282. 1978.
- [KUN80] Kunt, M.: "Traitement Numérique des Signaux". Editions Goergi, 1980.
- [KUN82] Kung, H.T.: "Why Systolic Architectures". IEEE Comp., pp.37-46. January, 1982.
- [LAL85] Lala, P.K.: "Fault Tolerant & Fault Testable Hardware Design". Prentice Hall Int., 1985.
- [LEC71] Lechner, R.J.: "Harmonic Analysis of switching Functions", publicado en "Recent Developments in Switching Theory", ed. Mukhopadhyay et al., Academic Press, 1971.
- [LEC85] Lechner, R.J.; Moezzi, A.: "Synthesis of Encoded PLAs", in "Spectral Techniques and Fault Detection", M.G. Karpovsky ed., pp.35-90, 1985.
- [LEV78] Levendel, Y; Breuer, M.A.: "Mathematical Properties of Boolean Transformations". Proc. IEEE 8th ISMVL; pp. 163-170.
- [LIN83] Lin, S.; Costello, D.J.Jr.: "Error Control Coding: Fundamentals and Applications". Prentice Hall. 1983.
- [LIN88] Lin, D.J., Bose, B.: "Theory and Design of t-Error Correcting and d(d>t)-Unidirectional Error Detecting (t-EC d-UED) Codes. IEEE Transactions on Computers, Vol. 37, No. 4, pp. 433-439, April 1988.
- [LIU87] Liu, D.L.; McCluskey, E.J.: "Designing CMOS Circuits for Switch-Level Testability". IEEE Design & Test of Comp., pp.42-49. August, 1987.
- [LIU88] Liu, D.L.; McCuskey, E.J.: "Design of large embedded CMOS PLA's for Built-in-Self Test". IEEE Trans. on Comp. Aid. Des., Vol7, pp.50-59. January, 1988.
- [LO87] Lo, C.-Y.; Nham, H.N.; Bose, A.K.: "Algorithms for an Advanced Fault Simulation System in MOTIS". IEEE Transactions on CAD, Vol. CAD-6, pp. 232-240, March 1987.
- [LOS78] Losq, J.: "Efficiency of Random Compact Testing". IEEE Transactions on Computers, Vol. C-27, No. 6, pp. 516-525, June 1978.
- [MAA87] Maamari, F., Rajski, J.: "Reconvergent Fanout Analysis and Fault Simulation Complexity of Combinational Circuits". Electronics Letters, Vol. 23, No. 21, pp. 1131-1133, October 1987.

- [MAA88a] Maamari, F., Rajski, J.: "A Fault Simulation Method Based on Stem Regions". IEEE FTCS-18, pp. 170-173. Junio 1988.
- [MAA88b] Maamari, F.; Rajski, J.: "A Reconvergent Fanout Analysis for Efficient Exact Fault Simulation of Combinational Circuits". IEEE FTCS-18, pp. 122-127. Junio 1988.
- [MAL84] Malaiya, Y.K.; Yang, S.: "The coverage problem for random testing". IEEE Int. Test Conf, pp. 237-245, 1984.
- [MAL85] Mallela, S.; Wu, S.: "A Sequential Circuit Test Generation System". Proc. Int. Test Conf., pp. 57-61. November, 1975.
- [MAN84] Mangir, T.E.: "Sources of failures and yield improvement for VLSI and restructurable interconnections for RVLSI and WSI: Part I". Proc. IEEE, vol. 72, pp. 690-704. June, 1984.
- [MCL81] McCluskey, E.J.; Bozorgui-Nesbat, S.: "Design for Autonomous Test". IEEE Transactions on Circuits and Systems, Vol. CAS-28, No. 11, pp. 1070-1078, November 1981.
- [MCL84] McCluskey, E.J.: "Verification Testing - A Pseudoexhaustive Test Technique". IEEE Transactions on Computers, Vol. C-33, No. 6, pp. 541-546, June 1984.
- [MCL85a] McCluskey, E.J.: "Built-In Self-Test Techniques". IEEE Design & Test, pp. 21-36, April 1985.
- [MCL85b] McCluskey, E.J.: "Built-In Self-Test Structures". IEEE Design & Test, pp. 29-36, April 1985.
- [MCL88] McCluskey, E.J.; Makar, S.; Mourad, S.; Wagner, K.D.: "Probability Models for Pseudorandom Test Sequences". IEEE Comp. Aid. Des., vol. 7, No. 1, pp. 68-74. January, 1988.
- [MEI74] Mei, K.C.Y.: "Bridging and Stuck-At Faults". IEEE Transactions on Computers, Vol. TC-23, No. 7, pp. 720-727, July 1974.
- [MEN78] Menon, P.R.; Chappell, S.G.: "Deductive Fault Simulation With Functional Blocks". IEEE Transactions on Computers, Vol. C-27, pp. 689-695, August 1978.
- [MET53] Metropolis N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A.; Teller, E.: "Equation of State Calculations by Fast Computing Machines". J. Chem. Phys., 21(6), pp. 1087-1092. June, 1953.
- [MIL81] Miller, D.M.: "Spectral Symmetry Tests". Proc. 11th IEEE ISMVL, pp. 130-134, 1981.
- [MIL83] Miller, D.M.; Muzio, J.C.: "Spectral Fault Signatures for Internally Unate Combinational Networks". IEEE Trans. on Comp., C-33. 1984.
- [MIL84] Miller, D.M.; Muzio, J.C.: "Spectral Fault Signatures for Single Stuck-At Faults in Combinational Networks". IEEE Transactions on Computers, Vol. TC-33, No. 8, pp. 765-769. 1984.
- [MIL85] Miller, D.M.; Muzio, J.C.: "Spectral Techniques for Fault Detection in Combinational Logic", in "Spectral Techniques and Fault Detection", M.G. Karpovsky ed., pp. 371-420, 1985.
- [MIN88] Min, Y.; Li, J.: "Strongly Fault Secure PLA's and Totally Self-Checking Checkers". IEEE Transactions on Computers, Vol. 37, No. 7, pp. 863-867, July 1988.
- [MIY87] Miyakawa, M.; Stojmenovic, I.; Lau, D.; Rosenberg, I.G.: "Classifications and basis enumerations in many-valued logics - a survey -". Proc. 17th IEEE ISMVL, pp. 152-160, 1987.
- [MON90] Montgomery, B.L.; Kumar, B.V.K.V.: "Systematic Random Error Correcting and All Unidirectional Error Detecting Codes". IEEE Trans. on Comp., Vol. 39, No. 6. June, 1990, pp. 836-840.
- [MOO85] Moore, W.R.: "A Review of Fault-Tolerant Techniques for the Enhancement of Integrated Circuit Yield". Proceedings of the IEEE, Vol. 74, No. 5, pp. 684-698. 1985
- [MOR85] Moraga, C.; Seseke, K.: "The Chrestenson Transform in Pattern Analysis", in "Spectral Techniques and Fault Detection", M.G. Karpovsky ed., pp. 143-178. 1985.
- [MUK70] Mukhopadhyay, A.; Schmitz, G.: "Minimization of Exclusive OR and logical equivalence of switching circuits". IEEE Trans. Comp., Vol. C-19, pp. 132-140, 1970.

- [MUT76] Muth, P.: "A Nine-Valued Circuit Model for Test Generation". IEEE Transactions on Computers, Vol. C-25, pp. 630-636, June 1976.
- [MUZ82] Muzio, J.C.; Miller, D.M.: "Spectral Techniques for Fault Detection". Proc. 12th Int. Symp. on Fault Tolerant-Computing, pp.297-302. 1982.
- [NAN87] Nanya, T.; Kawamura, T.: "A Note on Strongly Fault-Secure Sequential Circuits". IEEE Transactions on Computers, Vol. C-36, No. 9, pp. 1121-1128, September 1987.
- [NAN89] Nania, T.; Goosen, H.A.: "The Byzantine Hardware Fault Model". IEEE Trans. on Comp.-Aid. Des., Vol.8, No.11. November, 1989.
- [ORT89] Ortega, J.; Prieto, A.; Pelayo, F.J.; Lloris, A.: "Simulador de defectos físicos en PLAs a nivel de transistor". V Jornadas de Diseño Lógico, La Rábida (Huelva). Diciembre, 1989.
- [OTT88] Otten, R.H.J.M.; Ginneken, L.P.P.P. van: "Stop Criteria in Simulated Annealing". IEEE Int. Conf.Comp.Des., pp.549-552, 1988.
- [PAR85] Parker, J.P.; McCluskey, E.J.: "Analysis of Logic Circuits with Faults Using Input Signal Probabilities". IEEE Transactions on Computers, vol. 34; pp. 573-578, May, 1985.
- [PAR75] Parker, K.P.; McCluskey, E.J.: "Probabilistic Treatment of General Combinational Networks". IEEE Transactions on Computers, Vol. C-24, pp. 668-670, June 1975.
- [PET72] Peterson, W.W.; Weldon, E.J.Jr.: "Error-Correcting Codes". MIT Press, 1972.
- [PHE84] Phee, D.H.: "Probabilidad y Estadística, vol.3: Inferencia Estadística". Ed. Limusa, 1984
- [PIE87] Piestrak, S.J.: "Design of Fast Self-Testing Checkers for a Class of Berger Codes". IEEE Transactions on Computers, Vol. C-36, No. 5, pp. 629-634, May 1987.
- [PRA80] Pradhan, D.K.: "A New Class of Error-Correcting/Detecting Codes for Fault-Tolerant Computer Applications". IEEE Transactions on Computers, Vol. C-29, No. 6, pp. 471-481, June 1980.
- [PRA90] Pradham, D.K.; Gupta, S.K.; Karpovsky, M.G.: "Aliasing Probability for Multiple Input Signature Analyzer". IEEE Trans. on Comp., vol39, No.4, pp.586-591. April, 1990.
- [RAJ87] Rajski, J., Agarwal, V.K.: "Testing and Applications of Inverter-Free PLAS". IEEE Design & Test of Computers, pp. 30-40, December 1987.
- [RAS82] Rasmussen, R.A.: "Automated Testing of LSI". IEEE Computer, pp. 69-78, March 1982.
- [RED72] Reddy, S.M.: "Easily testable realizations for logic functions". IEEE Trans. Comp., vol.C-21, pp.1183-1188. November, 1972.
- [RED84] Reddy, S.M.; Agrawal, V.D.; Jain, S.K.: "A Gate Level Model for CMOS Combinational Logic Circuits with Application to Fault Detection". Proc. 21st. Design Automation Conference, pp. 504-509, June 1984.
- [RED85] Reddy, M.K.; Reddy, S.M.; Agrawal, P.: "Transistor Level Test Generation for MOS Circuits". Proc. 22nd Design Automation Conf., pp. 825-828, June 1985.
- [RED86] Reddy, M.K.; Reddy, S.M.: "Detecting FET Stuck-open Faults in CMOS Latches and Flip-Flops". IEEE Design & Test of Comp., vol.4, pp.17-26. October, 1986.
- [RED87] Reddy, S.M., Ha, D.S.: "A New Approach to the Design of Testable PLA's". IEEE Transactions on Computers, Vol. C-36, No. 2, pp. 201-211, February 1987.
- [REI88] O'Reilly, W.P.: "Ingeniería Electrónica asistida por Computador". Ed. Paraninfo. 1988.
- [ROB82] Robinson, J.P.; Yeh, C.-L.: "A method for modulo-2 minimization". IEEE Trans. on Comp., vol.C-31, pp.800-801, 1982.
- [ROB84] Roberts, M.W.; Lala, P.K.: "An algorithm for the partitioning of logic circuits". IEE Proceedings, Vol. 131, Pt. E, No. 4, pp. 113-118, July 1984.
- [ROB87] Robinson, J.P.; Saxena, N.R.: "A unified view of test compression methods". IEEE Trans. on Comp., vol.C-36, No.1. January, 1987.

- [ROT67] Roth, J.P.; Bouricius, W.C.; Schneider, P.R.: "Programmed Algorithms to Compute Tests to Detect and Distinguish between Failures in Logic Circuits". IEEE Transactions on Electronic Computers, Vol. EC-16, No. 5, pp. 567-580, October 1967.
- [RUI89] Ruiz, G.; Michell, J.A.; Burdon, A.: "Herramientas de ayuda a la diagnosis de fallos físicos en circuitos secuenciales NMOS/CMOS via análisis espectral". pp. 333-340.
- [SAH80] Sahni, S.; Bhatt, A.: "The Complexity of Design Automation Problems". Proc. 17th Design Autom. Control, Min., MN, pp.402-411, 1980.
- [SAL83] Saluja, K.K., Kinoshita, K., Fujiwara, H.: "An Easily Testable Design of Programmable Logic Arrays for Multiple Faults". IEEE Transactions on Computers, Vol. C-32, No. 11, pp. 1038-1046, November 1983.
- [SAL88] Saluja, K.K.; Sharma, R.; Kime, C.R.: "A concurrent Testing Technique for Digital Circuits". IEEE Transactions on Computer-Aided Design, Vol. 7, no. 12, pp. 1250-1259, December 1988.
- [SAM86] Sami, M.; Stefanelli, R.: "Reconfigurable Architectures for VLSI Processing Arrays". Proceedings of the IEEE, Vol. 74, No. 5, pp. 712-722, May 1986.
- [SAS81] Sasao, T.: "Multiple-Valued Decomposition of Generalized Boolean Functions and the Complexity of Programmable Logic Arrays". IEEE Transactions on Computers, Vol. C-30, No. 9, pp. 635-643, September 1981.
- [SAS86] Sasao, T.; Besslich, P.W.: "On the complexity of PLAs with EXOR arrays", IECEJ Tech. Rep. FTS 86-17. November, 1986.
- [SAS87] Sasao, T.: "Bounds on the Average Number of Products in the Minimal Sum-of-Products Expressions for Multiple-Valued Input Two-Valued Output Functions". Proc. 17th IEEE ISMVL; pp.260-267; 1987.
- [SAS88] Sastry, S.; Breuer, M.: "Detectability of CMOS Stuck-Open Faults Using Random and Pseudorandom Test Sequences". IEEE Transactions on Computer-Aided Design, Vol. 7, No. 9, pp. 933-997, September 1988.
- [SAS90] Sasao, T.; Besslich, P.: "On the Complexity of Mod-2 Sum PLAs". IEEE Trans. on Comp., Vol.39, No.2, pp.262-266. February, 1990.
- [SAV80] Savir, J.: "Syndrome-Testable Design of Combinational Circuits". IEEE Transactions on Computers, Vol. C-29, No. 6, pp. 442-451, June 1980.
- [SAV81] Savir, J.: "Syndrome-Testing of "Syndrome-Untestable" Combinational Circuits". IEEE Transactions on Computers, Vol. C-30, No. 8, pp. 606-608, August 1981.
- [SAV83] Savir, J.: "Good Controlability and Observability Do not Guarantee Good Testability". IEEE Trans. on Comp., Vol.C-32, No.12, pp.1198-1200. December, 1983.
- [SAV84a] Savir, J.; Ditlow, G.S.; Bardell, P.H.: "Random Pattern Testability". IEEE Transactions on Computers, Vol. C-33, No. 1, pp. 79-90, January 1984.
- [SAV84b] Savir, J.; Bardell, P.H.: "On Random Pattern Length". IEEE Trans. on Comp., vol.C-33, No.6, pp.467-474. June, 1984.
- [SAV87] Savir, J., McAnney, W.H., Vecchio, S.R.: "Fault Propagation Through Embedded Multiport Memories". IEEE Transactions on Computers, Vol. C-36, No. 5, pp. 592-602, May 1987.
- [SAY86] Sayers, I.L.; Kinniment, D.J.; Chester, E.G.: "Design of a reliable and self-testing VLSI datapath using residue coding techniques". IEE Proceedings, vol. 133, Pt. E, No. 3, pp. 169-179, May 1986.
- [SCH72] Schertz, D.R.; Metze, G.: "A New Representation for Faults in Combinational Circuits". IEEE Transactions on Computers, Vol. C-21, pp. 858-866, August 1972.
- [SCH75] Schnurmann, H.D.; Lindbloom, E.; Carpenter, R.G.: "The Weighted Random Test-Pattern Generator". IEEE Transactions on Computers, Vol. C-24, No. 7, pp. 695-700, July 1975.
- [SCH84] Schuster, M.D.; Bryant, R.E.: "Concurrent Fault Simulation of MOS Digital Circuits". Proc. MIT Conf. on Adv. Research in VLSI, pp. 129-138, January 1984.

- [SCH88] Schulz, M., Trischler, E., Sarfert, T.M.: "SOCRATES: A Highly Efficient Automatic Test Pattern Generation System". IEEE Transactions on Computer-Aided Design, Vol. 7, No. 1, pp. 126-137, January 1988.
- [SEL68] Seller, F.F.; Hsiao, M.Y.; Bearnson, L.W.: "Analyzing Errors with Boolean Difference". IEEE Trans. on Comp., Vol. C-17, pp. 676-683, July, 1968.
- [SER86] Serra, M.; Muzio, J.C.: "Testability by Sum of Syndromes". Proceedings of Technical Workshop: New Directions for IC Testing, March 18-20 1986, Victoria B.C. Canada, pp. 11-1-11-20.
- [SET84] Seth, S.C.; Agrawal, V.D.: "Characterizing the LSI Yield Equation from Wafer Test Data". IEEE Transactions on CAD, Vol. CAD-3, pp. 123-126, April 1984.
- [SET85a] Seth, S.C.; Pan, L.; Agrawal, V.D.: "PREDICT-Probabilistic Estimation of Digital Circuit Testability". Fault-Tolerant Comp. Symp. Digest, pp. 220-225, June 1985.
- [SET85b] Seth, S.C.; Agrawal, V.D.: "Cutting Chip Testing Costs". IEEE Spectrum, Vol. 22, pp. 38-45, April 1985.
- [SET90] Seth, S.C.; Agrawal, V.D.; Farhat, H.: "A Statistical Theory of Digital Circuit Testability". IEEE Trans. on Comp., vol. 39, No. 4, pp. 582-584, April, 1990.
- [SHA85] Upadhyaya, S.J.; Saluja, K.K.: "Signature Techniques in Fault Detection and Location", in "Spectral Techniques and Fault Detection", M.G. Karpovsky ed., pp. 421-476, 1985.
- [SHE75] Shedletsky, J.J.; McCluskey, E.J.: "The Error Latency of a Fault in a Combinational Digital Circuit". Proceedings of the Fault-Tolerant Computing Symposium, pp. 210-214, 1975.
- [SHE84] Shen, J.P.; Ferguson, F.J.: "The design of Easily Testable VLSI Array Multipliers". IEEE Trans. on Comp., vol. C-33, No. 6, June, 1984.
- [SHE85] Shen, J.P.; Maly, W.; Ferguson, F.J.: "Inductive Fault Analysis of MOS Integrated Circuits". IEEE Design & Test of Computers, Vol. 2, pp. 13-26, December 1985.
- [SHI82] Shiozaki, A.: "Single Asymmetric Error-Correcting Cyclic AN Codes". IEEE Transactions on Computers, Vol. C-31, No. 6, pp. 554-559, June 1982.
- [SHI84] Shin, K.G.; Lee, Y.-H.: "Error Detection Process-Model, Design, and Its Impact on Computer Performance". IEEE Transactions on Computers, Vol. C-33, No. 6, pp. 529-540, June 1984.
- [SIN84] Singer, D.M.: "Testability Analysis of MOS VLSI Circuits". Proc. Int. Test Conf., pp. 690-696, November 1984.
- [SMI79] Smith, J.E.: "Detection of Faults in Programmable Logic Arrays". IEEE Transactions on Computers, Vol. C-28, No. 11, pp. 845-853, November 1979.
- [SMI80] Smith, J.E.: "Measures of the Effectiveness of Fault Signature Analysis". IEEE Transactions on Computers, Vol. C-29, No. 6, pp. 510-514, June 1980.
- [SNE77] Snethen, T.J.: "Simulator-Oriented Fault Test Generator". Proc. 14th Design Automation Conf., pp. 88-93, June 1977.
- [SON80] Son, K.; Pradham, D.K.: "Design of programmable logic arrays for testability". Proc. IEEE Test. Conf., pp. 163-166, 1980.
- [SON85] Son, K.: "Fault Simulation with the Parallel Value List Algorithm". VLSI Systems Design, pp. 36-43, December 1985.
- [SOU78] Sousa, P.T. De; Mahtur, F.P.: "Shift-Out Modular Redundancy", IEEE Trans. Comp., C-27, pp. 624-627, 1978.
- [STA75] Stapper, C.H.: "On a composite model to the IC yield problems". IEEE J. Solid-State Circuits, vol. SC-10, pp. 537-539, December, 1975.
- [STA83] Stapper, Ch.H., Armstrong, F.M., Saji, K.: "Integrated Circuit Yield Statistics". Proceedings of the IEEE, Vol. 71, No. 4, pp. 453-470, April 1983.
- [SUK80] Suk, D.S., Reddy, S.M.: "Test Procedures for a Class of Pattern-Sensitive Faults in Semiconductors Random-Access Memories". IEEE Transactions on Computers, Vol. C-29, No. 6, pp. 419-428, June 1980.
- [SUS83] Susskind, A.K.: "Testing by Verifying Walsh Coefficients". IEEE Transactions on Computers, Vol. C-2, No. 2, pp. 198-201, February 1983.
- [TAM52] Tamari, D.: "Some mutual applications of logic and mathematics". Proc. 2nd Int. Colloq. of Mathematical Logic, pp. 89-90, August, 1952.

- [TAK87] Takamatsu, Y.; Kinoshita, K.: "CONT: A Concurrent Test Generation Algorithm". FTCS 17, pp.22-27. July, 1987.
- [TAM84] Tamir, Y.; Sequin, C.H.: "Design and Application of Self-Testing Comparators Implemented with MOS PLA's". IEEE Transactions on Computers, Vol. C-33, No. 6, pp. 493-506, June 1984.
- [TAN83] Tang, D.T., Woo, L.S.: "Exhaustive Test Pattern Generation with Constant Weight Vectors". IEEE Transactions on Computers, Vol. C-32, No. 12, pp. 1145-1150, December 1983.
- [TAN84] Tang, D.T., Chen, Ch.: "Logic Test Pattern Generation Using Linear Codes". IEEE Transactions on Computers, Vol. C-33, No. 9, pp. 845-849, September 1984.
- [TAO88] Tao, D.L.; Hartmann, C.R.P.; Lala, P.K.: "An Efficient Class of Unidirectional Error Detecting/Correcting Codes". IEEE Transactions on Computers, Vol. 37, No. 7, pp. 879-882, July 1988.
- [TEW89] Tewksbury, S.K., Hornak, L.A.: "Wafer Level System Integration: A Review". IEEE Circuits and Devices Magazine, pp. 22-30, September 1989.
- [THA80] Thatte, S.M.; Abraham, J.A.: "Test Generation for Microprocessors". IEEE Transactions on Computers, Vol. C-29, pp. 429-441, June 1980.
- [THA82] Thatte, S.M.; Ho, D.S.; Yuan, H.T.; Sridhar, T.: "An Architecture for Testable VLSI Processors". Proceedings of the International Test Conference, pp. 484-492, 1982.
- [THE81] Thévenod-Fosse, P.; David, R.: "Random Testing of the Data Processing Section of a Microprocessor". Proceedings of the Fault-Tolerant Computing Symposium, pp. 275-280, 1981.
- [THO71] Thomas, J.J.: "Automated Diagnostic Test Program for Digital Networks". Computer Design, pp.63-67. August, 1977.
- [THO75a] Thompson, E.W.; Szygenda, S.A.: "Digital Logic Simulation in a Time-Based, Table-Driven Environment, Part 2 Parallel Fault Simulation". Computer, Vol. 8, pp. 38-49, March 1975.
- [THO75b] Thompson, E.W.; Szygenda, S.A.: "Parallel Fault Simulation". IEEE Computer, pp. 38-44, March 1975.
- [TIM82a] Timoc, C.; Favennec, J.M.; Le Blanche, C.: "A Testable Regular Design". Proceedings of the International Conference on Circuits and Computers, pp. 210-213, 1982.
- [TIM82b] Timoc, C.; Stott, F.; Hess, L.: "A Novel Approach to Test Generation for VLSI". Proceedings of COMPCON S'82, pp. 78-86, 1982.
- [TIM83a] Timoc, C.; Buehler, M.; Griswold, T.; Pina, C.; Stott, F.; Hess, L.: "Logical Models of Physical Failures". Proceedings of the International Test Conference, pp. 546-553, 1983.
- [TIM83b] Timoc, C.; Scott, F.; Wickman, K.; Hess, L.: "Adaptative Self-test for a Microprocessor". IEEE Proc. Int. Test Conf, pp.701-703. 1983.
- [TOK79] Tokmen, V.H.: "Some properties of spectra of ternary logic functions". Proc. IEEE 9th ISMVL, 1979; pp. 88-93.
- [TOK80] Tokmen, V.H.: "Disjoint Descomposability of Multi-Valued Functions by Spectral Means". Proc. 10th ISMVL, pp.88-93, 1980.
- [TRA85] Trachtenberg, E.A.; Karpovsky, M.G.: "Filtering in a Communication Channel by Fourier Transforms over Finite Groups", in "Spectral Techniques and Fault Detection", M.G.Karpovsky ed., pp.179-216, 1985.
- [TRA87] Tran, A.: "Graphical method for the conversion of minterms to Reed-Muller coefficients and the minimisation of exclusiv-OR switching functions". IEEE Proceedings, Vol. 134, Pt. E, No. 2, pp. 93-99, March 1987.
- [TRE85] Treuer, R., Fujiwara, H., Agarwal, V.K.: "Implementing a Built-In Self-Test PLA Design". IEEE Design & Test, pp. 37-48, April 1985.
- [TRE87] Treuer, R., Agarwal, V.K., Fujiwara, H.: "A New Built-In Self-Test Design for PLA's with High Fault Coverage and Low Overhead". IEEE Transactions on Computers, Vol. C-36, No. 3, pp. 369-373, March 1987.
- [TSI89] Tsikuris, A.G.; Reklaitis, G.V.; Tenorio, M.F.: "Nonlinear Optimization Using Generalized Hopfield Networks". Neural Computation, Vol.1, No.4. 1989.
- [TSU87] Tsui, F.F.: "LSI/VLSI Testability Design". Ed. McGraw Hill. 1987.

- [TUN88] Tung, C.-H.; Robinson, J.P.: "A Fast Algorithm for Optimum Syndrome Space Compression". IEEE Transactions on Computers, Vol. 37, No. 2, pp. 228-232, February 1988.
- [ULR69] Ulrich, E.G.: "Exclusive Simulation of Activity in Digital Networks". Communications of the ACM, Vol. 12, pp. 102-110, February 1969.
- [ULR73] Ulrich, E.G.; Baker, T.: "The Concurrent Simulation of Nearly Identical Digital Networks". Proceedings of the Design Automation Workshop, pp. 145-150, 1973.
- [ULR74] Ulrich, E.; Baker, T.: "Concurrent Simulation of Nearly Identical Digital Networks". IEEE Computer, Vol. 7, pp. 39-44, April 1974.
- [VOG78] Vogel, A.: "Multiple bridging faults in monotone networks". Computers and Digital Techniques, Vol. 1, No. 2, pp. 49-52, May 1978.
- [WAD78a] Wadsack, R.L.: "Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits". Bell System Tech. Jour., Vol. 57, pp. 1449-1474, May-June, 1978.
- [WAD78b] Wadsack, R.L.: "Technology Dependent Logic Faults". Proceedings of COMPCON S'78, pp. 124-129, 1978.
- [WAG87] Wagner, K.D.; Chin, G.K.; McCluskey, E.J.: "Pseudorandom Testing". IEEE Trans. on Comp., vol. C-36, No. 3, pp. 332-343. March, 1987.
- [WAI85] Waicukauski, J.A.; Eichbelger, E.B.; Forlenza, D.O.; Lindbloom, E.; McCarthy, T.: "A statistical calculation of fault detection probabilities by Fast Fault Simulation". Proc. Int. Test Conf., pp. 779-784. November, 1985.
- [WAL88] Walczak, K.: "Deductive Fault Simulation for Sequential Module Circuits". IEEE Transactions on Computers, Vol. 37, No. 2, pp. 237-239, February 1988.
- [WAN80] Wang, S.L.; Avizienis, A.: "The design of totally self checking circuits using programmable logic arrays". Proc. Int. Symp. FTCS, pp. 173-180, 1979.
- [WAN88] Wang, L.-T.; McCluskey, E.J.: "Hybrid Designs Generating Maximum-Length Sequences". IEEE Transactions on Computer-Aided Design, Vol. 7, No. 1, pp. 91-99, January 1988.
- [WEY87] Wey, Ch.; Vai, M.; Lombarde, F.: "On the Design of a Redundant Programmable Logic Array (RPLA)". IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 1, pp. 114-117, February 1987.
- [WIL73] Williams, N.J.Y.; Angell, J.B.: "Enhancing Testability of Large-Scale Integrated Circuits via Test Points and Additional Logic". IEEE Transactions on Computers, pp. 46-60, January 1973.
- [WIL77] Williams, T.W.; Eichelberger, E.B.: "Random Patterns within a Structured Sequential Logic Design". Proceedings of the International Test Conference, pp. 19-26, 1977.
- [WIL81] Williams, T.W.; Brown, N.C.: "Defect Level as a Function of Fault Coverage". IEEE Transactions on Computers, Vol. C-30, pp. 987-988, December 1981.
- [WIL83] Williams, T.W.; Parker, K.P.: "Design for Testability A Survey". Proc. IEEE, Vol. 71, pp. 98-112, January 1983.
- [WIL88] Williams, T.W.; Daehn, W.; Gruetzner, M.; Starke, C.W.: "Bounds and Analysis of Aliasing Errors in Linear Feedback Shift Registers". IEEE Transactions on Computer Aided Design, Vol. 7, No. 1, pp. 75-83, January 1988.
- [WIL89] Williams, T.W.; Daehn, A.: "Aliasing errors in multiple signature analysis registers", in Proc. BIST Workshop, 1989.
- [YOU89] You, Y.; Hayes, J.P.: "Implementation of VLSI Self-Testing by Regularization". IEEE Transactions on Computer-Aided Design, Vol. 8, No. 1, pp. 1261-1271, January 1989.

DILIGENCIA:

Reunido el Tribunal examinador en el día de la fecha, constituido por:

- D. Pedro Cartujo Estebanez
- D. Angel M^o Buron Romero
- D. Elena Valderrama Valles
- D. Juan A. Mitchell Martin
- D. Franisco J. Pelayo Valle

para juzgar la Tesis Doctoral del Licenciado Don

Julio Ortega Lopera
se acordó por Unanimidad otorgar la calificación de: Opto Cum Laude

y para que conste, se extiende firmada por los componentes del Tribunal, la presente diligencia.

Granada, a 23 de Noviembre 1990

El Secretario,

El Presidente,

[Signature]
Ed.: Pedro Cartujo

El Vocal,

[Signature]

FDO: ANGEL M. BURON ROMERO

El Vocal,

[Signature]
FDO: JUAN A. MITCHELL

El Vocal,

[Signature]

Fdo: ELENA VALDERRAMA VALLES