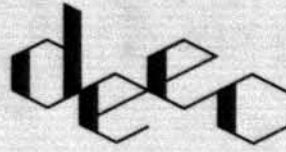


Bej:



DEPARTAMENTO DE
ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

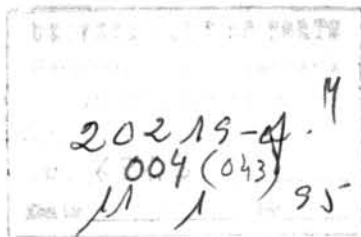
Os Algoritmos Adaptativos
no Diagnóstico de Falhas em
Cartas de Circuito Impresso
Equipadas com BST

FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Rua dos Bragas, 4099 Porto Codex - PORTUGAL

RA 753

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Departamento de Engenharia Electrotécnica e de Computadores



N.º. 34572

Os Algoritmos Adaptativos
no Diagnóstico de Falhas em
Cartas de Circuito Impresso
Equipadas com BST

António José Duarte Araújo

Licenciado em Engenharia Electrotécnica e de Computadores
pela Faculdade de Engenharia da Universidade do Porto

Dissertação submetida para satisfação parcial dos requisitos
do programa do curso de
Mestrado em Engenharia Electrotécnica e de Computadores
(área de especialização em Informática Industrial)

PORTO, Abril 1994

043M
A687a
2x. 3

Dissertação realizada sob a supervisão do
Professor Doutor José Alfredo Ribeiro da Silva Matos,
Professor Associado do
Departamento de Engenharia Electrotécnica e de Computadores da
Faculdade de Engenharia da Universidade do Porto

Resumo

A capacidade de controlar e observar componentes seleccionados em cartas de circuito impressos são aspectos determinantes para a detecção e diagnóstico de faltas. A utilização do *boundary scan test* possibilita elevados índices de controlabilidade e observabilidade, contribuindo assim para aumentar a capacidade de diagnóstico.

A geração de testes para CCI's pode ser feita por algoritmos que possuem diferentes compromissos relativamente ao número de vectores de teste gerados e à capacidade de diagnóstico. Neste trabalho faz-se uma descrição funcional dos vários algoritmos existentes para este fim e expõem-se conclusões acerca das suas características. Em situações exigentes de diagnóstico, os algoritmos que apresentam melhor desempenho são os algoritmos adaptativos que realizam o teste em dois passos.

A implementação de um algoritmo para utilização num equipamento de teste automático implica numa fase prévia a sua especificação formal. A principal contribuição desta tese reside nesta especificação, descrevendo-se as fases de geração de vectores de teste, detecção e análise de síndromas para os algoritmos adaptativos descritos.

Abstract

The observability and controllability of selected components in printed circuit boards are important aspects for the detection and diagnosis of faults in electronic circuits. The use of the boundary scan test methodology allows a high degree of controllability and observability, thus contributing for high degrees of fault detection and diagnosis.

The generation of test vectors can be made by algorithms that implement different compromises with respect to the number of test vectors and diagnosis capability. This work presents a functional description of several existing algorithms, along with conclusions regarding their characteristics. In applications requiring diagnosis, the most efficient are the adaptative algorithms that execute a test in two steps.

The implementation of an algorithm to be used in an automatic test equipment requires, in a previous phase, its formal specification. The main contribution of this thesis is the formal specification of these algorithms. The main aspects addressed are the test vector generation, fault detection and analysis of syndromes for the adaptative algorithms presented here.

Agradecimentos

Desejo aqui expressar o meu agradecimento ao Professor José Alfredo Ribeiro da Silva Matos, meu orientador científico, pela sua permanente disponibilidade em proporcionar as melhores condições de trabalho e pela constante confiança manifestada no decorrer deste trabalho.

Agradeço a todos os meus colegas do grupo CAD e Microelectrónica do INESC pelo bom ambiente de trabalho e pela permanente disposição em ajudar.

Finalmente, desejo agradecer à minha família pela paciência e apoio demonstrados.

*À Mariazinha
e aos meus pais*

Índice

1	Introdução	1
2	O teste de CCI	3
2.1	Noções básicas	3
2.2	O ciclo de vida de uma CCI	5
2.3	Falhas existentes em CCI	7
2.3.1	Falhas dos componentes	7
2.3.2	Falhas de fabrico	9
2.3.3	Falhas de desempenho	10
2.3.4	Influência da tecnologia das cartas	10
2.4	Tecnologias de teste tradicionais	12
2.4.1	O teste in-circuit	12
2.4.2	O teste funcional	13
2.4.3	Outras possibilidades	15
2.4.4	Descrição sumária	16
2.5	Tecnologias emergentes	17
2.6	Tecnologia BST	19
2.6.1	Topologia de um circuito BST	19
2.6.2	Instruções de teste	20
2.6.3	Teste de CCI com BST	21
2.6.4	Limitações e vantagens	21
2.7	Conclusão	23
3	Detecção e diagnóstico de faltas em CCI	24
3.1	Noções básicas	24
3.2	Modelo de faltas	27
3.3	Algoritmos de geração de vectores de teste	27
3.3.1	Algoritmos de um passo	28

3.3.1.1	Algoritmo da pesquisa binária normal	28
3.3.1.2	Algoritmo da pesquisa binária modificado	29
3.3.1.3	Algoritmo da pesquisa binária modificado com complemento	30
3.3.1.4	Algoritmo da sequência caminhante	32
3.3.1.5	Algoritmo do peso mínimo	33
3.3.1.6	Algoritmo da independência máxima	37
3.3.1.7	Algoritmo do auto-diagnóstico	40
3.3.1.8	Algoritmo do diagnóstico global	42
3.3.1.9	Algoritmo do diagnóstico máximo	44
3.3.2	Algoritmos adaptativos	50
3.3.2.1	Algoritmo de teste dos W vectores	50
3.3.2.2	Algoritmo do vector único	52
3.3.2.3	Algoritmo dos C vectores óptimos	53
3.3.2.4	Algoritmo do diagnóstico em dois passos	56
3.3.2.5	Algoritmo adaptativo A1	58
3.3.2.6	Algoritmo adaptativo A2	60
3.4	Comentários finais	62
4	Análise comparativa dos algoritmos	63
4.1	Número de vectores de teste	63
4.2	Capacidade de diagnóstico	66
4.3	Conclusão	67
5	Implementação dos algoritmos adaptativos	68
5.1	Introdução	68
5.2	Procedimento de teste e diagnóstico	69
5.2.1	Geração do primeiro conjunto de vectores de teste	71
5.2.2	Detecção de faltas	73
5.2.3	Geração do segundo conjunto de vectores de teste	75
5.2.4	Diagnóstico de faltas	77
5.3	Alteração ao algoritmo dos C vectores óptimos	78
5.3.1	Análise do segundo passo do algoritmo	78
5.3.2	Proposta de uma solução	80
5.3.3	Descrição algorítmica	80
5.4	Conclusão	85
6	Conclusão	86

A Apêndice	88
A.1 Independência diagonal dum conjunto de vectores	88
A.2 VTPs gerados pelo algoritmo da independência máxima	90
A.3 Geração dum sequência caminhante	93
Bibliografia	95

Lista de Figuras

2.1	Fases e etapas do ciclo de vida de uma CCI.	5
2.2	A etapa do teste de produção.	6
2.3	Influência da taxa de falhas dos componentes nas falhas das CCIs.	9
2.4	Magnitude das causas dos defeitos nos vários tipos de CCIs.	11
2.5	Constituição básica de um circuito BST.	20
3.1	A localização de vectores de teste em ligações.	25
3.2	Situação em que ocorre uma síndrome do tipo <i>aliasing</i>	26
3.3	Situação em que ocorre uma síndrome do tipo <i>confounding</i>	26
3.4	Quantidade de VTSs possíveis em função do peso e do número de <i>bits</i>	35
3.5	Faltas não diagnosticáveis.	45
3.6	Curto-circuito não identificado por uma sequência diagonalmente independente.	45
3.7	Circuito aberto não identificado por uma sequência diagonalmente independente.	46
3.8	Curto-circuito não identificado por um conjunto de vectores independente.	46
3.9	Grafo de vizinhanças e atribuição de cores.	49
3.10	Situação com faltas múltiplas.	59
4.1	As funções $p = f(N)$ em algoritmos de um só passo.	65
4.2	As funções $p = f(N)$ em algoritmos adaptativos.	66
5.1	Metodologia de diagnóstico usando um algoritmo adaptativo.	70

Lista de Tabelas

2.1	Utilização dos testes a componentes.	8
2.2	Distribuição de falhas típica em CCIs com 175 componentes.	8
3.1	Aplicação do algoritmo da pesquisa binária normal.	29
3.2	Aplicação do algoritmo da pesquisa binária modificado.	30
3.3	Aplicação do algoritmo da pesquisa binária modificado com complemento.	31
3.4	Aplicação do algoritmo da sequência caminhante.	33
3.5	Aplicação do algoritmo do peso mínimo.	36
3.6	Aplicação do algoritmo da independência máxima.	40
3.7	Aplicação do algoritmo do auto-diagnóstico.	42
3.8	Aplicação do algoritmo do diagnóstico global.	43
3.9	Aplicação do algoritmo do diagnóstico máximo.	49
3.10	Aplicação do algoritmo de teste dos W vectores.	51
3.11	Aplicação do algoritmo do vector único.	53
3.12	Aplicação do algoritmo dos C vectores óptimos.	55
3.13	Aplicação do algoritmo do diagnóstico em dois passos.	57
3.14	Aplicação do algoritmo adaptativo A1.	59
3.15	Aplicação do algoritmo adaptativo A2.	61
4.1	Características quantitativas e qualitativas dos algoritmos apresentados.	64
5.1	Eliminação se síndromas do tipo <i>confounding</i> pelo algoritmo dos C vectores óptimos.	79
A.1	Características dos VTSs gerados pelo algoritmo da independência máxima.	90

Capítulo 1

Introdução

As necessidades crescentes de dispor de novas ferramentas de teste que proporcionem bons resultados em curtos intervalos de tempo exige a exploração das capacidades de metodologias de teste existentes para que possam constituir soluções efectivas e praticáveis. Uma área em que nesse sentido tem sido feito algum investimento é a do diagnóstico de falhas em cartas de circuito impresso, tema em que se insere este trabalho.

Esta tese encontra-se estruturada em cinco capítulos para além deste capítulo introdutório.

O objectivo do capítulo 2 é o de situar o leitor na área do teste de falhas em cartas de circuito impresso. Começa por fazer uma apresentação dos conceitos mais frequentemente utilizados ao longo deste trabalho, após a qual se caracterizam as falhas vulgarmente encontradas numa carta de circuito impresso (CCI). Inclui-se uma panorâmica sobre as tecnologias de teste mais enraizadas na indústria fazendo-se a devida caracterização. As incapacidades reveladas por estas são ultrapassadas pela tecnologia *Boundary Scan Test* (BST), a qual permite ultrapassar as dificuldades mais importantes no teste ao nível da carta, realçadas no capítulo anterior. Sobre BST apresenta-se um resumo da sua implementação, revelam-se as limitações e vantagens mais significativas, e refere-se a importância do teste de ligações numa CCI pela taxa de ocorrências deste tipo de falhas.

O capítulo 3 apresenta ao leitor algoritmos para a geração de vectores destinados ao teste das ligações eléctricas em cartas de circuito impresso. O processo adoptado para a sua exposição consiste numa descrição exaustiva e detalhada de cada um dos algoritmos, o que se justifica pelo facto de assim ser possível entender com clareza as diferenças de desempenho existentes entre eles.

No capítulo 4 reúnem-se várias conclusões importantes acerca das características dos algoritmos apresentados no capítulo anterior, através duma análise comparativa, sobretudo

fundamentada na dimensão do conjunto de vectores de teste necessários para a obtenção de certos níveis de detecção e diagnóstico de falhas. O objectivo desta detalhada análise é a eleição de um conjunto de algoritmos que reúnem as melhores características de comportamento.

A classe de algoritmos evidenciada no capítulo anterior conduz ao capítulo 5. Neste capítulo é apresentada uma metodologia procedimental de diagnóstico que permite decompor um algoritmo de teste e diagnóstico segundo várias fases de implementação. Deste modo, descrevem-se os referidos algoritmos utilizando uma pseudo-linguagem de programação, e apresenta-se uma inovação relativamente a um dos algoritmos descritos.

Finalmente, o capítulo 6 é constituído por um conjunto de conclusões que encerram em si o que de mais importante foi escrito e desenvolvido. São ainda perspectivadas algumas direcções futuras de possíveis desenvolvimentos.

Ao longo deste trabalho utiliza-se o estilo *itálico* para realçar o local onde é feita pela primeira vez referência a um conceito que é exposto, *inclinado* para indicar termos numa língua estrangeira e **sans serif** para escrever as palavras chave da pseudo-linguagem de programação utilizada no capítulo 5.

Capítulo 2

O teste de CCI's

O teste de sistemas complexos baseados em componentes VLSI é uma actividade onde aspectos como qualidade, tempo, custo e disponibilidade de equipamento são importantes. Tradicionalmente, a escolha do teste ao nível da carta consistia simplesmente na selecção de uma ou mais fases de teste a aplicar [DDD⁺93]; por exemplo, teste *in-circuit* seguido de teste funcional. Contudo, se é requerido um certo nível de qualidade, ou se o acesso à carta é dificultado pela utilização de tecnologia de montagem superficial, torna-se claro que a tradicional estratégia de teste deixa de ser adequada.

A primeira secção deste capítulo destina-se a familiarizar o leitor com alguns conceitos, frequentemente utilizados ao longo desta tese, relacionados com o teste de CCI's. Na secção que se lhe segue são apresentadas as várias fases constituintes do ciclo de vida de uma CCI, desde a sua especificação até à sua utilização, realçando a importância do teste ao longo do ciclo. Depois, na terceira secção, é feita uma caracterização das falhas que mais frequentemente ocorrem. Na quarta secção são expostas as tecnologias de teste ditas tradicionais, realçando-se as respectivas virtudes e limitações. Na quinta secção referem-se tendências de evolução da tecnologia dos componentes utilizados em CCI's, essencialmente ao nível do encapsulamento e montagem, e a consequente ineficácia crescente das tecnologias de teste descritas. Por fim, a sexta secção introduz o *Boundary Scan Test* como medida de resolução dos principais problemas colocados às técnicas de teste descritas. Salienta-se ainda na mesma secção a importância do teste de ligações das CCI's.

2.1 Noções básicas

O teste de um sistema consiste em exercitá-lo e analisar a resposta resultante para verificar se funciona correctamente [ABF90]. Esta tarefa permite efectuar a *deteccção*

de falhas, mas nem sempre permite fazer o *diagnóstico* das mesmas, isto é, localizar e identificar as falhas ocorridas. O teste de uma CCI, independentemente da fase em que é efectuado, tem como objectivo comum determinar se a carta funciona devidamente, ou se pelo contrário, possui defeitos. Entenda-se por *defeito*, ou *falha*, a causa física que está na origem do mau funcionamento dum carta. Na secção 2.3 referem-se os defeitos que mais frequentemente ocorrem em CCIs.

Uma importante questão que se põe quando se efectua o teste dum sistema é a avaliação do próprio teste [Bat85], [ABF90], que consiste em determinar a *efectividade*, ou qualidade, dum dada estratégia ou tipo de teste. Designa-se por *estratégia de teste* a selecção de alternativas de projecto e etapas de teste que permitam atingir a mais elevada qualidade do produto com o menor custo [DDD⁺93]. A avaliação do teste é feita usualmente no contexto dum *modelo de faltas*, que constitui uma forma mais conveniente de representar as falhas físicas, sendo medida pela razão entre o número de faltas que o teste consegue detectar e o número de faltas que podem ocorrer no universo de faltas considerado. Esta razão é normalmente referida como sendo a taxa de *cobertura de faltas* do teste.

Os custos relacionados com o processo de teste de uma CCI estão directamente relacionados com a complexidade do teste, dependendo esta da complexidade da carta. Um modo de limitar estes custos é orientar o projecto da CCI, e dos respectivos componentes, no sentido de ter em conta a testabilidade do produto final. Esta metodologia de projecto é correntemente designada por *projecto orientado para a testabilidade*, ou DFT (*design for testability*). Em [WP83] e [ABF90] são descritas várias técnicas de DFT, ao nível de CIs e CCIs, que traduzem esforços empregues especificamente para tornarem os dispositivos testáveis. Dois importantes atributos intimamente relacionados com a testabilidade são a controlabilidade e a observabilidade [ABF90]. A *controlabilidade* é a capacidade de estabelecer o valor dum sinal específico em cada nó dum circuito através da atribuição de valores nas entradas do circuito. A *observabilidade* é a capacidade de determinar o valor dum sinal em qualquer nó dum circuito, controlando as suas entradas e observando as respectivas saídas. Uma consequência directa da controlabilidade dum circuito é a *resolução* do teste e diagnóstico que se consegue atingir, ou seja, a precisão na localização e identificação de falhas possibilitada pela capacidade de isolamento eléctrico de elementos do circuito. Em [Gol79] é apresentado um método de análise de circuitos digitais em termos de funções que caracterizam a controlabilidade e a observabilidade de circuitos combinacionais e sequenciais, através dum medição quantitativa da dificuldade em controlar e observar os valores lógicos em nós internos dum circuito, a partir da consideração da sua topologia, sem analisar sequências particulares de vectores de teste. As várias técnicas de DFT existentes têm como objectivo comum alterar o projecto de um circuito no sentido

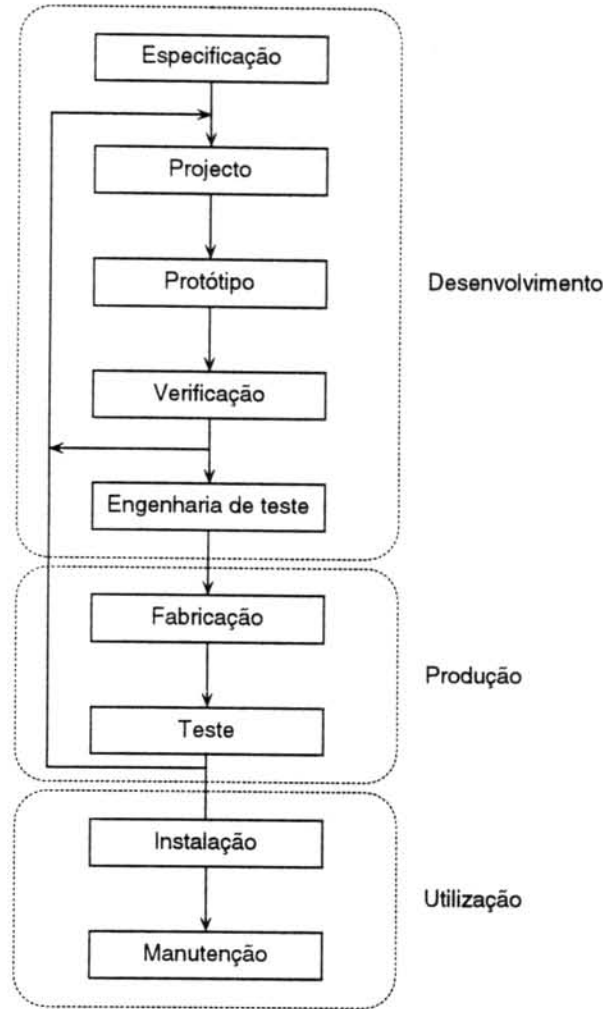


Figura 2.1: Fases e etapas do ciclo de vida de uma CCI.

de maximizar os seus níveis de controlabilidade e observabilidade.

2.2 O ciclo de vida de uma CCI

O ciclo de vida de qualquer produto é constituído por três fases: desenvolvimento, produção e utilização. Em particular, uma CCI também possui este ciclo de vida. Cada uma destas fases requer alguma forma de teste [Ben82], com ou sem reparação de falhas encontradas, conforme se ilustra na figura 2.1, e onde se apresentam também as várias etapas de cada fase [MBD91], [DDD+93].

A primeira etapa do desenvolvimento é naturalmente a especificação da carta, a seguir

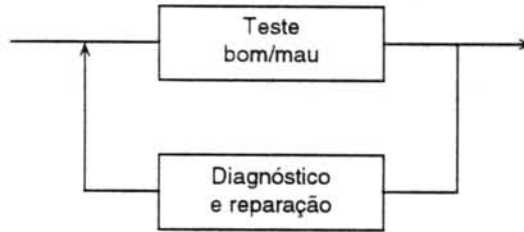


Figura 2.2: A etapa do teste de produção.

à qual se efectua o seu projecto. A etapa que se lhe segue tem por objectivo construir algumas CCIs que servem de protótipos para verificação da especificação inicial do projecto. Geralmente, são necessárias algumas iterações no ciclo projecto-protótipo-verificação até um protótipo funcionar correctamente, conforme a especificação. A verificação é seguida da etapa de engenharia de teste, que é responsável pela preparação e geração de programas de teste, e programação do equipamento de teste a utilizar no teste de produção.

Concluído o desenvolvimento, segue-se a fase da produção, que inclui a fabricação e montagem das CCIs que são sujeitas ao teste de produção. Esta etapa da produção encontra-se esquematizada na figura 2.2. Em [MBD91] encontram-se descritos modelos, não só para o teste de produção, como também para os testes de instalação e manutenção. O *teste de produção* consiste essencialmente na detecção de falhas nas CCIs, por aplicação de estímulos de teste e recolha das respectivas respostas, que são comparadas com os valores esperados. Resulta daqui um *teste bom/mau*, que verifica se as cartas estão em bom estado ou se precisam de ser reparadas. Para isto é necessário diagnosticar as falhas, para que se possa proceder à sua eliminação. O teste de produção de CCIs garante a qualidade da montagem e permite monitorar o processo de fabrico, com o objectivo de minimizar os custos de diagnóstico e reparação de falhas nas fases posteriores de teste do sistema, teste final, ou teste de manutenção. Note-se que todas as CCIs reparadas devem ser submetidas a um novo teste, com o objectivo de detectar falhas que tenham permanecido, ou surgido durante o próprio processo de reparação [DDD⁺93]. Existem situações em que o custo do teste de produção não justifica que este se efectue, ocorrendo quando a produção é de pequeno volume, as cartas são demasiado simples, ou a percentagem destas sem falhas é elevada, isto é, geralmente superior a 80% [Bat85]. Os resultados conseguidos nesta etapa de teste reflectem-se no *rendimento da produção*, que se define como sendo, a razão entre o número de CCIs que passam o teste de produção, ou seja, que não contêm faltas detectáveis, e o número total de CCIs que são testadas [TL81], [Bat85].

A fase de utilização inicia-se com a instalação da CCI no sistema a que se destina,

tornando-o funcional conforme pretendido através do teste de instalação. Nesta fase, que se deseja a mais longa, será obviamente necessária alguma forma de manutenção para ultrapassar possíveis falhas futuras da CCI. Resulta daqui a necessidade do teste de manutenção.

Resumindo, durante a fase de desenvolvimento, o principal objectivo é validar o projecto lógico e a sua implementação, isto é, é dada ênfase à validação do projecto e não a provar-se a existência ou não de falhas. Quando o projecto atinge a linha de produção, o ênfase cai no teste de falhas, tornando a intervenção do teste nesta fase mais relevante devido à quantidade de CCIs envolvidas na produção [Ben82].

2.3 Falhas existentes em CCIs

Uma carta de circuito impresso, pode ser testada em dois estados distintos, correspondendo a situações em que a carta ainda não possui componentes, e em que a carta possui todos os componentes adquirindo a funcionalidade para que foi projectada. Nesta última situação, as falhas físicas que normalmente ocorrem na CCI podem dividir-se em três categorias [Bat85]: falhas dos componentes, causadas pela não verificação das especificações dos componentes; falhas de fabrico, devidas a erros surgidos na montagem da CCI e/ou na soldadura dos componentes; e falhas de desempenho, geralmente resultantes dum projecto débil ou de falhas dinâmicas dos componentes.

2.3.1 Falhas dos componentes

Entre 10 e 20% das falhas verificadas ao nível da CCI são potencialmente causadas por componentes defeituosos. Embora se possa minimizar esta taxa de falhas, seleccionando os componentes de modo a garantir o seu bom funcionamento a 100%, esta situação nem sempre será a melhor em termos económicos. Se por um lado é garantido o bom estado de todos os componentes a incluir na CCI, por outro lado esta tarefa traz custos adicionais que podem ser insuportáveis. Além disso, alguns fabricantes de componentes garantem uma qualidade e fiabilidade que não justificam o teste a 100% dos componentes. Outra filosofia que é adoptada por algumas companhias, tendo por fim estabelecer um equilíbrio entre os custos do teste de componentes e a minimização das falhas daqui resultantes, é o teste com base num nível aceitável de qualidade, segundo critérios definidos pelas próprias companhias. Existem ainda estratégias que combinam os três tipos de teste a componentes descritos. Assim, para uma mesma CCI, poderão coexistir componentes que são 100% testados, componentes que são sujeitos a um teste de aceitação, e componentes

que nem sequer são testados. A escolha dos tipos de teste a efectuar aos componentes tem em conta a origem dos componentes, a utilização a que se destinam, a fiabilidade e a tolerância que se pretendem, mas sobretudo factores económicos.

Na tabela 2.1 [Bat85] mostra-se como a indústria encara o teste a componentes que utiliza em CCIs, perante o seu tipo. É notória a diferença entre os tipos de teste efectuados a componentes VLSI e a componentes analógicos.

Componentes	Teste 100% (%)	Teste de aceitação (%)
VLSI	29	27
LSI	27	38
SSI/MSI	24	45
Analógicos	22	50

Tabela 2.1: Utilização dos testes a componentes.

A tabela 2.2 [Bat85] ilustra a distribuição típica das falhas em CCIs com 175 componentes para três taxas de defeitos destes. Note-se que a percentagem de CCIs com um dado número de falhas por CCI, é uma medida estatística do número médio de componentes considerados com falhas físicas. Por exemplo, para a taxa de 1% de componentes defeituosos, 17.8% das CCIs não apresentam nenhuma falha, 30.4% das CCIs apresentam em média 1 falha, 26.2% das CCIs apresentam em média 2 falhas, 15.4% das CCIs apresentam em média 3 falhas, 6.9% das CCIs apresentam em média 4 falhas, e apenas 3.4% das CCIs apresentam em média 5 ou mais falhas.

Falhas por CCI	Taxa de defeitos dos componentes		
	0.5%	1.0%	2.0%
0	41.8	17.8	3.3
1	36.3	30.4	11.3
2	15.9	26.2	18.6
3	4.7	15.4	21.1
4	1.1	6.9	18.4
5+	0.2	3.4	27.2

Tabela 2.2: Distribuição de falhas típica em CCIs com 175 componentes.

Considerando a taxa de 1% de componentes com falhas, a figura 2.3 ilustra a distribuição das falhas numa CCI em função do número de componentes desta, tomando

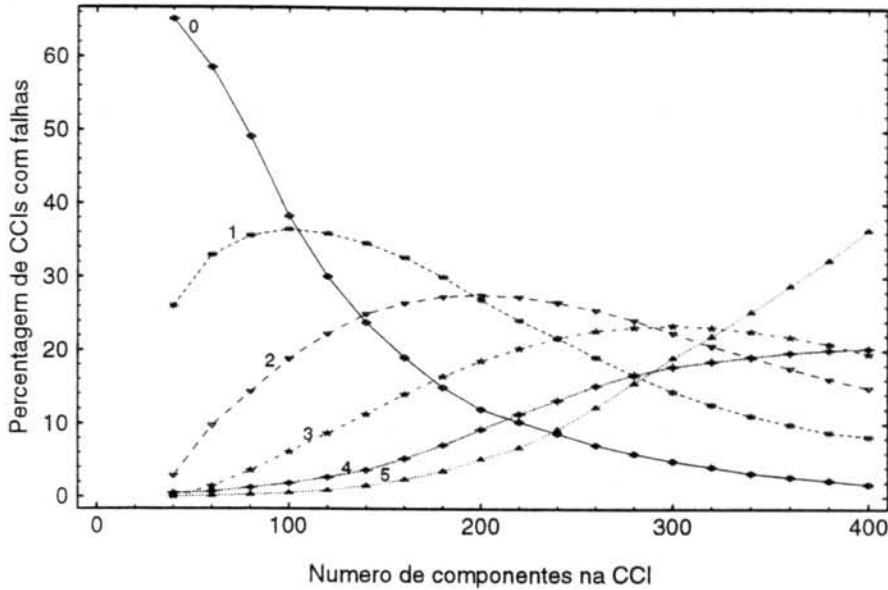


Figura 2.3: Influência da taxa de falhas dos componentes nas falhas das CCIs.

como parâmetro o número médio de falhas por CCI [Bat85], assinalado nas várias curvas. Repare-se que o que se disse no parágrafo anterior, a título de exemplo para a taxa de 1% de componentes com falhas, pode agora ser verificado pela figura 2.3.

2.3.2 Falhas de fabrico

A segunda categoria de falhas físicas que podem ocorrer numa CCI inclui as falhas originadas pelos processos de fabrico relacionados com a montagem das CCIs e soldadura dos componentes. As falhas de montagem consistem em erros de inserção de componentes, tais como, componentes errados, componentes que faltam, e componentes mal orientados; pinos de ligação dos componentes inclinados ou dobrados, quebrados, ou não existentes; e ligações erradas, quebradas, ou não existentes. As falhas de soldadura resultam em situações indesejadas de curto-circuito, causadas principalmente pelo fluxo de solda entre ligações ou pinos vizinhos, ou circuito aberto, causadas pela falta de solda, componentes mal fixados, ou ligações imperfeitas entre pinos e pistas da CCI através dos furos da carta.

Esta segunda categoria de falhas assume valores que se situam entre os 25 e os 55%, entre as causas de defeitos em CCIs. Contudo, podem obter-se valores significativamente mais reduzidos, entre 10 a 20%, utilizando equipamento de inserção automática de componentes com verificação, mantendo no entanto algum grau de intervenção manual, conforme os custos económicos daí resultantes.

2.3.3 Falhas de desempenho

A terceira categoria de falhas inclui as resultantes dum mau projecto, sendo tipicamente causadas por má interacção entre componentes, utilização imprópria das saídas, existência de ruído excessivo e sinais distorcidos. Estas falhas de desempenho podem também ser motivadas pela mistura pouco cuidada de componentes provenientes de vários fabricantes e pelas variações das especificações dos componentes.

2.3.4 Influência da tecnologia das cartas

As três categorias de falhas dos componentes que acabaram de ser descritas levam a diferentes taxas de falhas das CCIs, conforme o tipo de componentes utilizados. Na figura 2.4 podem ver-se os valores percentuais de falhas [Bat85], variando entre um valor mínimo e máximo, em CCIs analógicas, digitais e mistas, para as várias classes de falhas, onde se nota que os defeitos decorrentes do processo de fabrico são em maior número que os originados por falhas dos componentes e mau desempenho.

Em CCIs analógicas, os defeitos devidos aos processos de fabrico ocorrem em menor número que em CCIs digitais e mistas, devido a normalmente existirem menos componentes, ao maior afastamento entre eles, ao maior espaçamento entre as ligações das cartas, e ao menor número de pinos a soldar. O maior número de defeitos apresentado pelas CCIs digitais, deve-se essencialmente à grande densidade de componentes, pistas das cartas demasiado próximas, e elevado número de pinos a soldar por cada componente. Sendo assim, os valores esperados das percentagens para as CCIs mistas, situam-se entre os revelados para as cartas analógicas e cartas digitais, sendo aproximadamente proporcional à mistura de componentes analógicos e digitais.

No que respeita às falhas dos componentes, nota-se que estas têm maior impacto nas CCIs analógicas do que nas CCIs digitais, pois a maior parte dos fabricantes praticam algum nível de teste dos componentes digitais, efectuando com menor frequência o teste aos componentes analógicos. Este comportamento, onde é colocado mais ênfase nos componentes digitais do que nos analógicos, deve-se à complexidade e aos custos crescentes dos componentes digitais e à progressiva tendência para tornar as cartas puramente digitais. Utilizando tecnologia mista, os valores das falhas em CCIs situam-se, mais uma vez, entre os valores das cartas analógicas e das cartas digitais.

As falhas de desempenho são mais significativas em CCIs digitais do que em CCIs analógicas. Contudo, repare-se na figura 2.4, em cartas digitais o valor mínimo da percentagem destas falhas tende para zero, uma vez que se utilizam componentes SSI ou MSI, que têm provas dadas no que se refere a maturidade de desempenho. As CCIs mistas

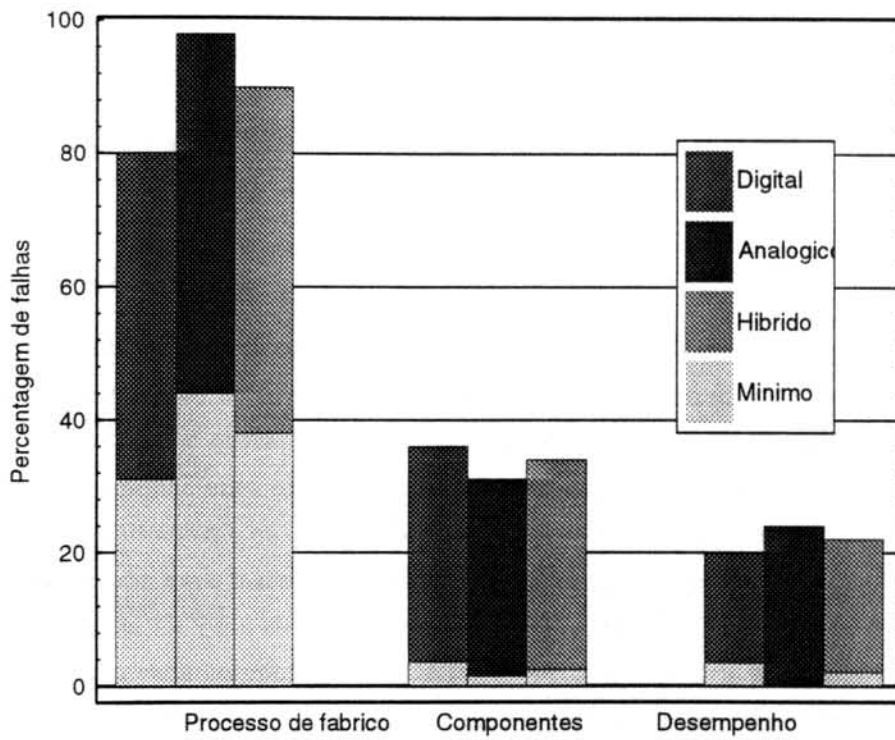


Figura 2.4: Magnitude das causas dos defeitos nos vários tipos de CCIs.

apresentam valores intermédios no que se refere a esta categoria de falhas.

2.4 Tecnologias de teste tradicionais

Um procedimento de teste de CCI caracteriza-se essencialmente pela tecnologia empregue na realização do teste. Ao equipamento que permite aplicar sinais de teste a uma dada CCI e recolher as respectivas respostas dá-se o nome de *testador*. Cada tipo de testador é genericamente designado por um Equipamento de Teste Automático, doravante referido por ETA. O sistema completo que inclui *software* de utilização, gestão e suporte do ETA, é conhecido por Sistema de Teste Automático (STA). Em [Ben82] encontra-se a descrição de alguns ETAs disponíveis comercialmente.

Descrevem-se de seguida as tecnologias de teste mais enraizadas na indústria, de acordo com a classificação feita em [Sch89], apontando-se as respectivas vantagens, desvantagens e requisitos.

2.4.1 O teste *in-circuit*

A estratégia do teste *in-circuit* consiste no teste individual dos componentes numa CCI, com os objectivos de detectar defeitos de conectividade, isto é, situações de circuito aberto e curto-circuito, e outras falhas do processo de fabrico, e de detectar falhas nos componentes. O equipamento de teste necessário para efectuar o teste *in-circuit* inclui, em particular, um mecanismo que proporciona o acesso mecânico e eléctrico à carta a testar (CAT), composto por uma matriz de agulhas que permite estabelecer contacto eléctrico entre cada ligação da CCI e o sistema de teste [Par87], [Sch89]. Os testes são efectuados separadamente para cada componente da CAT, e o facto de a matriz de agulhas permitir o acesso directo aos componentes da carta, conduz a uma elevada resolução do teste e diagnóstico.

Esta estratégia de teste manifesta características importantes:

- Não é necessário conhecer e entender a função a que se destina a CCI para preparar e executar o teste.
- O diagnóstico de falhas é particularmente simples e rápido, pois coincide praticamente com a detecção de falhas.
- As falhas múltiplas podem ser diagnosticáveis com um único passo de teste, pois à medida que os componentes vão sendo testados, vai sendo elaborada uma lista dos que apresentam falhas.

Para a sua execução ser possível, o teste *in-circuit* apresenta algumas exigências a vários níveis, conforme se indicam a seguir.

- A CAT deve possuir os seguintes requisitos:
 - Possibilidade de inserção num dispositivo onde é estabelecido vácuo para criar os contactos.
 - Posicionamento mecânico preciso através de furos de orientação.
 - Facilidades de teste proporcionadas pela tecnologia empregue.

- O sistema de teste deve possuir as seguintes características:
 - Capacidade de acesso aos componentes da CAT por meio de uma matriz de agulhas.
 - Velocidade de medida que permita a rápida execução de um grande número de testes digitais e analógicos.
 - Possibilidade de separar electricamente componentes ligados entre si, para permitir isolar os componentes a testar.
 - Instrumentação analógica rápida com possibilidade de interligação a um grande número de canais de teste.
 - Grande número de canais de teste digitais de controlo/observação, com capacidade de fornecer correntes de condução elevadas e memória para armazenamento de vectores de teste.

- O *software* de teste deve revelar os conhecimentos e capacidades dum operador de teste experimentado, incluindo nomeadamente:
 - Testes predefinidos para componentes analógicos.
 - Uma biblioteca de testes para componentes digitais.
 - Análise e descrição dos circuitos da CAT.
 - Rapidez na geração automática dos programas de teste.

2.4.2 O teste funcional

O teste funcional é outra abordagem que com os devidos investimentos na sua preparação e execução, também permite o diagnóstico da maioria das falhas que são causadas pelos componentes e processos de fabrico das CCI's. Esta estratégia de teste consiste em o

ETA emular o funcionamento eléctrico do sistema onde será inserida a CAT, aplicando sinais de entrada típicos, e medindo e avaliando os sinais de saída, recorrendo ao conector de entradas e saídas da CAT [Des88]. Resultam deste procedimento níveis de controlabilidade e observabilidade bastante limitados [Ray80], pois não existe acesso directo às ligações internas da carta. Contudo, durante o teste, a carta é colocada a funcionar duma maneira similar à que se pretende para o ambiente a que se destina [Par87], isto é, consegue-se recriar o funcionamento real da carta. No que respeita à preparação da sequência de teste, a carta é tratada como uma unidade indivisível. Assim, uma carta que contenha milhares de portas lógicas necessita de ser tratada como uma simples entidade, o que dificulta a tarefa de preparação do teste. A cobertura de falhas deste método [Sch89] depende fortemente da estrutura dos circuitos da CAT, da informação disponível sobre eles, e dos recursos disponíveis para a preparação do teste.

A execução do teste funcional assume verificados os requisitos que se apontam de seguida.

- As ferramentas de programação usadas no teste funcional devem facilitar a execução das seguintes acções:
 - Troca de informação detalhada com a fonte de projecto da CAT, acerca da sua função.
 - Simulação do funcionamento de circuitos digitais para a determinação automática dos respectivos vectores de teste.
- A CAT deve verificar os seguintes requisitos:
 - Documentação detalhada, especificação precisa da funcionalidade, e descrição exacta do fluxo de sinais.
 - Sinais de teste externos devem poder ser propagados através de todos os componentes do circuito.
- O sistema de teste funcional deve ser dotado de:
 - Uma *interface* universal para os conectores das cartas.
 - Estímulos de teste digitais e analógicos, e instrumentos de medida facilmente configuráveis.
 - Canais de controlo/observação para a geração/aquisição de vectores de dados.

- O *software* para o teste funcional deve ser caracterizado pelas seguintes possibilidades:
 - Descrição dos circuitos da CAT.
 - Análise do fluxo de sinais da CAT.
 - Conjunto de falhas da CAT programável, como meio de prova da eficácia das rotinas de diagnóstico.
 - Intercâmbio de informação com programas de simulação para a geração de teste digital.

2.4.3 Outras possibilidades

Combinando as técnicas de teste *in-circuit* e funcional, podem obter-se soluções interessantes para determinadas situações de teste, conjugando as vantagens de cada técnica.

O teste *combinado*, também designado frequentemente por *teste de agrupamentos*, consiste na aplicação conjunta dos testes *in-circuit* e funcional, com o objectivo de aproveitar as vantagens de ambas as técnicas de teste, e que se resumem de seguida.

- Teste estrutural da CAT rápido e pouco dispendioso relativamente à sua programação.
- Diagnóstico das falhas estruturais rápido sem recorrer a tempo adicional.
- Teste funcional detalhado de partes específicas da CAT, permitindo conhecer as respectivas funções.

A metodologia do teste combinado aproxima-se do teste *in-circuit*, pois é através de uma matriz de agulhas que se aplicam os sinais de teste funcional a componentes, ou a grupos de componentes, tornando assim possível conhecer o funcionamento da CAT. A óptima acessibilidade da técnica *in-circuit* a nós internos da CAT, e a possibilidade de isolar electricamente o agrupamento de componentes a testar, permite que a resolução do teste e diagnóstico conseguida por um testador combinado se situe entre a resolução conseguida pelo teste funcional e teste *in-circuit*.

Indicam-se de seguida as características mais marcantes desta forma de teste.

- Os requisitos para a CAT correspondem aos já apresentados para o teste *in-circuit*, entre os quais se apresentam como mais importantes, a existência de documentação detalhada sobre os blocos funcionais da CAT, e a especificação precisa da função e fluxo dos respectivos sinais.

- Um testador combinado é basicamente constituído pelo mesmo equipamento dum testador *in-circuit*, incluindo ainda estímulos de teste digitais e analógicos, e instrumentos de medida facilmente configuráveis, e fácil selecção das condições de sincronização entre canais de controlo/observação digitais.

O *teste de desempenho* é uma versão melhorada do teste funcional para CCIIs equipadas com microprocessadores [BA84], [LH88] e estruturas baseadas em barramentos. Embora o conceito de teste corresponda ao do teste funcional, dispõe de capacidades de teste e emulação orientadas para barramentos e componentes VLSI.

Indicam-se de seguida as características mais marcantes desta forma de teste.

- O *hardware* adicional caracteriza-se pela existência de unidades de emulação de microprocessadores específicos que permitem ao sistema de teste assumir o controlo da CAT, e unidades de teste para memórias que aumentam a velocidade de teste de RAMs e ROMs presentes na CAT.
- O *software* adicional caracteriza-se pela existência de códigos das instruções independentes dos microprocessadores, permitindo ao programador do teste reproduzir a actividade do microprocessador contido na CAT sem ter que o conhecer profundamente, e rotinas de teste normalizadas para simplificar os programas de teste correspondentes a grandes bancos de memória.

2.4.4 Descrição sumária

Apresenta-se de seguida um resumo das principais características das técnicas de teste *in-circuit* e funcional.

- Vantagens do teste *in-circuit*:
 1. Detecção e diagnóstico de falhas rápido.
 2. A geração dos programas de teste requer menor nível técnico e menos tempo para o seu desenvolvimento.
 3. Requer menor tempo para integrar o teste no sistema de produção.
 4. Capacidade de eliminar componentes que tenham passado o teste funcional.
 5. Capacidade de detectar várias falhas num único passo de teste.
 6. Boa relação custo/desempenho.

7. Baixo custo inicial e de manutenção do testador.

- Desvantagens do teste *in-circuit*:

1. Não testa o desempenho dum carta.
2. O teste só é efectuado a temperaturas ambientes.
3. Não detecta falhas de temporização dos sinais na carta.
4. Não testa a interacção dos circuitos.
5. O teste bom/mau é demorado.
6. O custo do equipamento, incluindo as pontas de prova responsáveis pelos contactos físicos entre este e a CAT, é elevado.

- Vantagens do teste funcional:

1. Testa o funcionamento da carta em condições reais.
2. Testa a interacção dos componentes.
3. Detecta falhas de temporização de sinais entre componentes.
4. Testa o desempenho da carta.
5. Teste bom/mau rápido.

- Desvantagens do teste funcional:

1. A geração dos programas de teste requer maiores níveis técnicos e mais tempo para o seu desenvolvimento.
2. Elevados custos de compra e manutenção do testador.
3. Diagnóstico de falhas lento e com baixa resolução.

2.5 Tecnologias emergentes

A grande evolução que se tem verificado nos últimos anos na tecnologia de fabrico de componentes electrónicos, em especial na tecnologia de circuitos integrados, tem levantado problemas crescentes às tradicionais técnicas de teste de CCI's. Referem-se de seguida alguns exemplos de tecnologias que implicam novas direcções no desenvolvimento de tecnologias de teste capazes de enfrentar os problemas causados pela complexidade dos componentes e da sua miniaturização.

A crescente utilização da tecnologia de montagem superficial tem sido o principal entrave à técnica de teste *in-circuit* [Dem94], devido ao reduzido espaçamento entre os

pinos dos CIs e à existência de componentes em ambos os lados das CCI. Por outro lado, a grande complexidade dos circuitos torna muito difícil a geração de testes funcionais. Além da tecnologia de montagem superficial, duas outras tecnologias de montagem tornam impraticáveis o teste pelos moldes habituais: a tecnologia *tape-automated bonding*, TAB, e *flip-chip bonding*, FCB. Na primeira, são estabelecidas ligações eléctricas entre o núcleo do CI e uma película onde vários CIs ficam depositados para posterior utilização por fixação nas CCI através dos terminais existentes na fita de polímero. A tecnologia FCB, mais recente, consiste na fixação do substrato, onde está implementado o CI, na CCI, ou noutro substrato, através de soldadura ou um material adesivo condutor [Mis93], [LB93]. A delicadeza desta tecnologia reside no facto de os *pads* do núcleo do CI ficarem voltados directamente para a superfície de fixação.

Também a grande proliferação de circuitos integrados de aplicação específica, comumente designados por ASICs, dificultam a realização do teste segundo as técnicas tradicionais, pela sua natureza específica do ponto de vista funcional.

Um dos avanços que se espera vir a ser dos mais significativos na indústria electrónica [Mis93] reside no encapsulamento de circuitos integrados e nas tecnologias de interligação subjacentes. A necessidade cada vez maior de miniaturização do encapsulamento de circuitos integrados, imposta pelo aumento de funcionalidade e desempenho destes, está a ter reflexos importantes no desenvolvimento dos chamados módulos de circuitos múltiplos, vulgarmente designados por MCM (*multichip module*). Um *módulo de circuitos múltiplos* é uma estrutura que consiste no agrupamento de dois ou mais CIs interligados entre si, suportados por um substrato comum, e encapsulados como uma unidade. O encapsulamento de circuitos múltiplos permite atingir circuitos de elevada densidade, com reduzidos comprimentos das interligações, e melhor desempenho do módulo resultante. Em [LB93] é apresentada uma interessante perspectiva global sobre MCM referindo, nomeadamente, capacidades, tamanho, velocidade, aplicações, materiais, processos de fabrico, e custo. Actualmente, considerando apenas circuitos encapsulados individualmente, a área de uma CCI ocupada por silício num sistema típico é inferior a 10% da área total. Contudo, espera-se atingir brevemente uma densidade efectiva da ordem dos 50% [Her93], através duma maior aproximação física dos CIs, quer a nível bidimensional com os MCMs, quer a nível de montagem vertical por meio de técnicas 3D. Estas exigências de miniaturização dos encapsulamentos implicam que possam efectuar-se interligações físicas com muito pequena espessura para que possam tornar-se efectivas no seu objectivo de aumentar a percentagem de silício numa CCI. Num MCM, a densidade de interligação de CIs pode ir até aos 90% do substrato, enquanto que numa CCI convencional este valor não vai além dos 10% [LB93]. Num MCM é possível atingirem-se interligações eléctricas com o comprimento de

4m por cm^2 de substrato dedicado a interligações [Car93].

Aos aspectos apontados, acrescenta-se o facto de as próprias CCIs terem vindo a tornar-se cada vez mais complexas, pois cada vez são mais as camadas de metalização que as constituem, diminuindo assim a acessibilidade a certas ligações.

A principal conclusão que se retira do que acabou de se expor é a necessidade de uma técnica de teste de CCIs que consiga ultrapassar os principais problemas colocados pelas tecnologias descritas. Na sequência disto apresenta-se na secção seguinte a tecnologia BST.

2.6 Tecnologia BST

A criação da norma IEEE 1149.1 — “IEEE Standard Test Access Port and Boundary Scan Architecture” [IEE90] deveu-se às dificuldades crescentes de aplicação das tecnologias de teste anteriormente apresentadas, definindo como projectar e incluir capacidades de testabilidade em CIs digitais [MT92] que simplificarão o teste. Estas capacidades podem ser usadas no teste de CIs, no teste de CCIs que usam componentes compatíveis com a norma e ainda ao nível do sistema, num sentido mais lato, formado por um conjunto de cartas. Surgida em Fevereiro de 1990, esta norma tem tido desde então uma aceitação crescente por parte dos fabricantes de CIs [MBD91], [Mau92], [Mau93], [Gre93], [Dan93] e [Teg93]. Para além de [IEE90] muita outra literatura existe sobre a tecnologia BST. Vejam-se por exemplo [Com90], [MT90], [BO91], [MT91], [Par92] e [WE93].

2.6.1 Topologia de um circuito BST

O BST é uma técnica de teste que envolve CIs projectados com registos de deslocamento colocados entre cada pino do CI e a lógica interna situada no núcleo. A figura 2.5 permite visualizar a configuração de um CI com BST.

Cada registo de deslocamento designa-se por célula BST. Estas células permitem controlar e observar o que acontece em cada pino da entrada e em cada pino da saída, forçando e capturando, respectivamente, valores lógicos nos referidos pinos dos CIs. A interligação das células BST forma uma cadeia de registos, a que se dá o nome de *Boundary Register* (BR). Os circuitos BS têm um porto de acesso dedicado, chamado *Test Access Port* (TAP), que permite a entrada de sinais para um controlador, e as já referidas células BST. A funcionalidade destas é comandada pelo controlador do TAP. Na figura 2.5 são visíveis os sinais referentes à lógica BS que constituem o TAP, sendo constituído por quatro pinos obrigatórios (TDI, TDO, TCK, TMS) e um pino opcional (TRST).

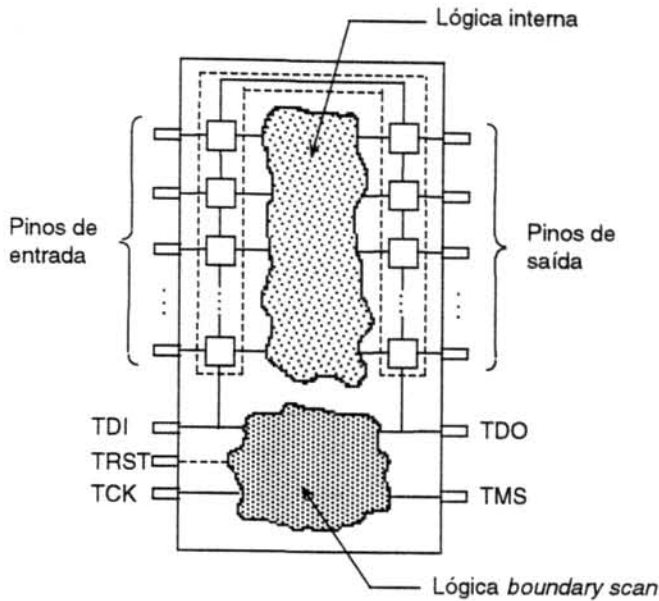


Figura 2.5: Constituição básica de um circuito BST.

- *Test Data In* (TDI) – entrada série para dados de teste e *bits* de instruções.
- *Test Data Out* (TDO) – saída série para os dados de teste.
- *Test Clock* (TCK) – relógio independente que serve a componente BST do circuito.
- *Test Mode Select* (TMS) – entrada que permite o controlador do TAP mudar de estado em estado.
- *Test Reset* (TRST) – entrada opcional que permite inicializar assincronamente a lógica de teste.

2.6.2 Instruções de teste

Além do BR, existem outros registos num componente BS, como o registo de *bypass* que providencia um caminho de um *bit*, permitindo minimizar a distância entre a entrada TDI e a saída TDO, o registo de instrução que é carregado em série com os *bits* de cada instrução de modo a permitir ao circuito a realização de várias funções, ou registos de dados definidos pelo utilizador.

As instruções de teste existentes em todos os dispositivos BS são:

- **BYPASS** – selecciona o registo de *bypass* e coloca as células BS em modo transparente.

- EXTEST – possibilita o teste das ligações entre CIs, por aplicação de valores no início das ligações, através dos pinos de saída, e captura de valores no fim das ligações através dos pinos de entrada.
- BYPASS – permite recolher ou inicializar os valores presentes em todos os pinos da CCI simultaneamente.

Existem ainda quatro instruções de teste que são opcionais: INTEST, RUNBIST, ID-CODE e USERCODE. As duas primeiras facultam o teste interno dum CI por aplicação de estímulos externos ou recorrendo à lógica interna de auto-teste [McC85] do CI, respectivamente, e as duas últimas permitem identificar os CIs montados na carta.

2.6.3 Teste de CCIs com BST

O mais importante modo de teste do BST é o modo externo accionado pela instrução EXTEST. Esta facilidade de teste torna possível a verificação de todas as ligações numa CCI estabelecidas entre células BST. É condição necessária para a execução do teste externo que a infraestrutura BST esteja operacional, pois ela própria pode possuir faltas. Assim, antes de executar qualquer teste a ligações deve ser feito o teste de integridade da infraestrutura BST da CCI [dJvdH91]. Em CCIs completamente equipadas com CIs BST é possível obter um elevado grau de diagnóstico de falhas devido à acessibilidade a cada CI, tornada possível pela cadeia BST que percorre todos os CIs.

Um estudo apresentado em [Sch89] revela a distribuição típica de falhas físicas em CCIs, assumindo que todos os componentes usados no processo de montagem foram testados encontrando-se funcionais antes da montagem. De acordo com os resultados obtidos, a distribuição de falhas é:

- 5% de falhas dos componentes, devidas à operação de montagem.
- 30% de falhas de montagem, devidas a troca de componentes e má orientação deles.
- 65% de falhas das ligações, devidas à ocorrência de circuitos abertos e curto-circuitos.

Estes resultados salientam a grande importância que tem o teste de ligações numa CCI.

2.6.4 Limitações e vantagens

O BST tem algumas limitações inerentes à própria metodologia, sobretudo no que respeita à rapidez de realização de testes e à capacidade de detecção de faltas [Fer92]. O

deslocamento em série de estímulos de teste e de respostas é um processo que requer uma parcela significativa do tempo de preparação do teste, tornando esta metodologia desadequada à realização de testes com carácter funcional. Quanto às limitações na capacidade de detecção, elas devem-se ao facto de um curto-circuito entre duas saídas dum circuito ter frequentemente por resultado um valor analógico, que terá de ser lido por uma entrada digital (a célula BST), o que origina um grau de incerteza quanto à validade dos valores lidos.

A estas limitações acrescenta-se ainda o custo inerente à implementação de BST num CI, que consiste fundamentalmente em três factores [FMP91], [SM90]:

- quatro (ou cinco) pinos adicionais,
- área de silício adicional para a lógica BST,
- aumento do tempo de propagação, devido à lógica contida entre a entrada e a saída de cada célula BST, e
- consumo de corrente adicional devido à lógica BST.

O aumento do tempo de propagação é um factor que só em casos muito especiais constituirá uma desvantagem relevante, já que apresenta valores típicos inferiores a 1 ns. Mesmo assim, nos casos em que este atraso seja intolerável, é possível a utilização de uma outra configuração para as células BST, onde a eliminação do atraso de propagação é trocada por uma redução na funcionalidade [IEE90]. A área de silício necessária para implementar a lógica BST, por sua vez, tem pouco significado nos circuitos de grande densidade de integração (VLSI) que mais tipicamente utilizam esta metodologia de teste. Mesmo em alguns casos de circuitos de baixa densidade de integração a implementação de BST pode justificar-se por conferir melhores características de testabilidade nos sistemas em que estes componentes virão a ser incluídos.

Quanto às vantagens destacam-se as seguintes [Par89], [SM90]:

- Os índices de controlabilidade e observabilidade proporcionados pelo BST permitem o desenvolvimento de equipamentos de teste mais eficientes e mais económicos.
- Tempos de teste inferiores, sobretudo para as CCI's cujo teste só fosse possível por técnicas de teste funcional.
- Reutilização de testes. Os testes de produção de um CI, acessíveis através da sua infraestrutura BST, podem ser reutilizados com o CI inserido na CCI, e com a CCI inserida no sistema.

2.7 Conclusão

Este capítulo começou por introduzir um conjunto de conceitos necessários à exposição dos assuntos contidos na tese. Fez-se logo de seguida uma análise do ciclo de vida de uma CCI e uma caracterização das falhas físicas correntemente encontradas durante o teste. Com o objectivo de dar a conhecer tecnologias de teste, foram descritas as técnicas *in-circuit* e funcional. A incapacidade destas foi justificada pelos desenvolvimentos recentes das tecnologias de montagem e encapsulamento. Finalmente, foi exposta a recente tecnologia BST com o fim de proporcionar uma solução aos problemas que se levantam às primeiras técnicas de teste discutidas [Joh93]. Face ao conjunto de benefícios e limitações desta tecnologia, o sucesso do BST parece irreversível, em consequência das potencialidades que exhibe face aos desafios criados pelas possibilidades da microelectrónica e das novas tecnologias de encapsulamento e de montagem.

Capítulo 3

Detecção e diagnóstico de faltas em CCI

A necessidade de efectuar o teste de CCI pode dever-se a dois objectivos fundamentais: o teste bom/mau, onde apenas interessa apurar se existem ou não faltas, e a localização e identificação das mesmas quando é requerido um diagnóstico. Os vários algoritmos que possibilitam a detecção e/ou o diagnóstico de faltas são expostos neste capítulo.

Na primeira secção introduz-se a principal terminologia usada. Conceitos como vector de teste/resposta sequencial/paralelo e síndrome são introduzidos.

Com o objectivo de sistematizar a apresentação dos vários algoritmos incluídos, estes apresentam-se divididos em dois grandes grupos, e para cada um deles é fornecido um exemplo de aplicação.

3.1 Noções básicas

No teste de CCI podem considerar-se várias formas de vectores de teste de acordo com a abordagem descrita em [JY89] para a geração de vectores de teste, detecção e diagnóstico de faltas. Descreve-se a seguir a classificação assim adoptada:

- *Vector de teste paralelo (VTP)*: representa os valores lógicos aplicados em paralelo ao conjunto de todas as ligações da CCI.
- *Vector de resposta paralelo (VRP)*: representa os valores lógicos capturados no conjunto de todas as ligações da CCI, como resultado da aplicação dum VTP.
- *Vector de teste sequencial (VTS)*: corresponde ao conjunto de valores lógicos aplicados sequencialmente a uma ligação da CCI, resultante da aplicação duma sequência

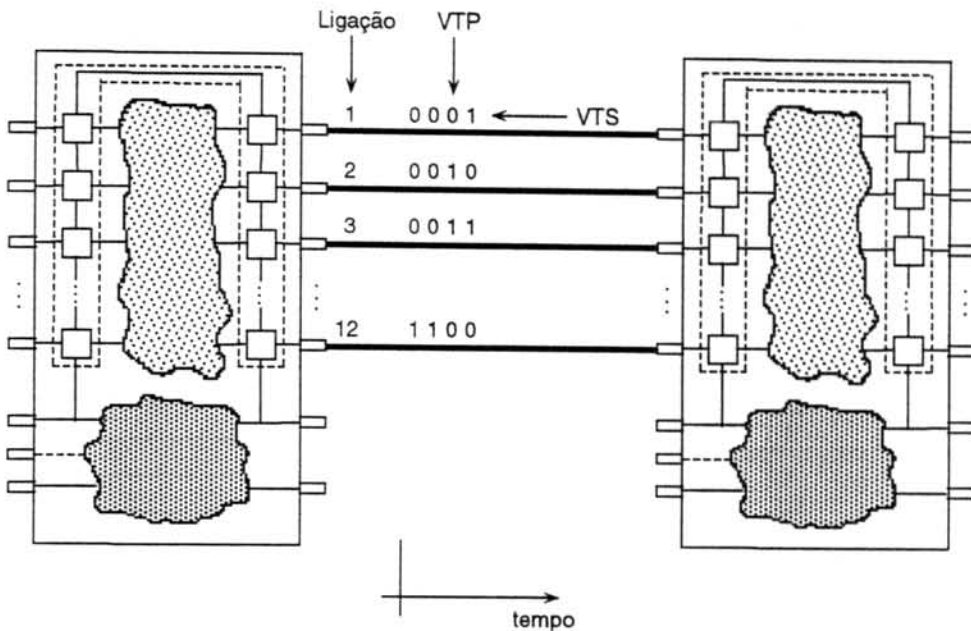


Figura 3.1: A localização de vetores de teste em ligações.

de VTPs.

- *Vector de resposta sequencial (VRS)*: corresponde ao conjunto de valores lógicos capturados sequencialmente numa ligação da CCI, como resultado da aplicação dum VTS.

A figura 3.1 localiza num conjunto de ligações os vetores de teste anteriormente classificados. Cada VTS, sendo único, pode ser considerado um identificador duma ligação. Assim, o conjunto de ligações duma CCI pode ser identificável pelo conjunto de VTSs. Para uma ligação sem qualquer tipo de falta o VRS é igual ao respectivo VTS.

Quando uma ligação numa CCI aparenta uma falta, ao respectivo VRS dá-se o nome de *síndrome* se o seu conteúdo for tal que possa levar a conclusões erradas sobre a existência da falta. Há a considerar dois tipos importantes de síndromas que ocorrem frequentemente [JY89], afectando a capacidade de diagnóstico de faltas de um dado conjunto de vetores de teste.

- *Síndrome do tipo aliasing*: ocorre quando, num conjunto de três ou mais ligações que apresentam um VRS comum e igual ao VTS atribuído a uma delas, a conjunção dos VTSs atribuídos às restantes ligações deste conjunto continua a originar o mesmo

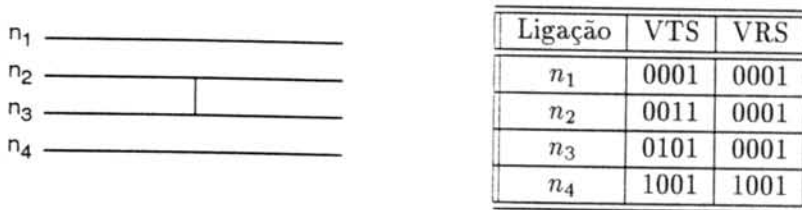


Figura 3.2: Situação em que ocorre uma síndrome do tipo *aliasing*.

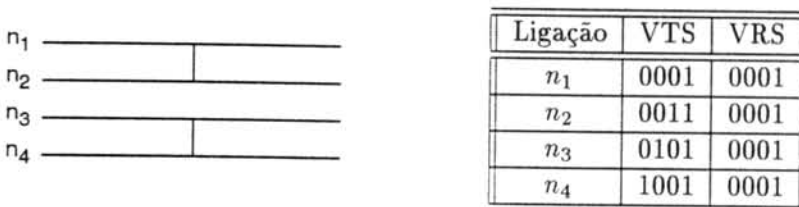


Figura 3.3: Situação em que ocorre uma síndrome do tipo *confounding*.

VRS, impossibilitando nestas circunstâncias distinguir se a ligação cujo VRS coincide com o respectivo VTS estará ou não envolvida no curto-circuito.

- *Síndrome do tipo confounding*: ocorre quando os VRSs produzidos por curto-circuitos independentes são iguais, tornando-se impossível distinguir se as ligações que apresentam este VRS comum estarão ou não todas curto-circuitadas entre si.

Em relação a este último tipo de síndrome, define-se ainda o *grau de confounding* como sendo o número máximo de faltas independentes que podem existir num conjunto de ligações que respondem com a mesma síndrome do tipo *confounding*. Nas figuras 3.2 e 3.3 ilustram-se situações que dão origem a síndromas dos tipos *aliasing* e *confounding*, respectivamente, para os VTSs indicados.

A causa da ocorrência de síndromas está nas características e propriedades dos vectores de teste e não no curto-circuito em si. Repare-se que um síndrome pode ser simultaneamente dos tipos *aliasing* e *confounding*, desde que se verifiquem simultaneamente as respectivas condições. Repare-se que enquanto a detecção de faltas não é afectada pela ocorrência de síndromas do tipo *aliasing* ou *confounding*, o diagnóstico de faltas apenas poderá ser completo quando não estiver presente nenhuma das síndromas. Este aspecto reflecte-se naturalmente na complexidade dum algoritmo usado para eliminar as situações de ambiguidade causadas por ambas as síndromas.

3.2 Modelo de faltas

Um modo de facilitar o tratamento da informação para efeitos de teste e diagnóstico consiste em estabelecer um modelo no qual as faltas físicas possam ser substituídas por faltas lógicas, de modo a traduzirem o efeito da falta física na operação do circuito [ABF90].

Um modelo de faltas deve permitir analisar facilmente o efeito das falhas físicas, sendo normalmente necessário estabelecer compromissos para equilibrar a complexidade do modelo de faltas necessária para uma determinada precisão com a facilidade da análise.

O modelo de faltas que aqui vai ser adoptado supõe o uso da tecnologia TTL que determina o comportamento lógico de um curto-circuito entre duas ligações. Assim, o modelo de faltas que será usado supõe:

- faltas do tipo *fixo-a* (*fixo-a-0* ou *fixo-a-1*),
- faltas do tipo circuito aberto, onde é assumido que uma entrada flutuante num circuito origina a captura do valor 1, e
- faltas do tipo curto-circuito, nas quais é assumido um comportamento lógico do tipo e.

Neste modelo não é considerada a ocorrência de faltas em simultâneo. Repare-se que o comportamento dos curto-circuitos depende da tecnologia, e que na tecnologia TTL o 0 tem “maior peso” que o 1, e daí o e lógico. Para uma outra tecnologia o modelo de faltas deve ser revisto, mantendo-se porém os aspectos essenciais da metodologia de teste e diagnóstico.

3.3 Algoritmos de geração de vectores de teste

Apresentam-se nesta secção vários algoritmos existentes para o teste de ligações em CCI. As diferenças existentes entre os algoritmos descritos e a respectiva complexidade do procedimento de geração de vectores de teste, resultam da estratégia adoptada para satisfazer uma solução de compromisso entre o número de VTPs e a capacidade de diagnóstico de faltas do tipo curto-circuito.

Na descrição de cada algoritmo são referidos aspectos determinantes para a avaliação do desempenho, tais como, número de VTPs gerados, capacidade de diagnóstico condicionada pela existência de síndromas e complexidade temporal [AHU74]. Deste modo, resulta uma abordagem sistemática, que permite ao leitor uma melhor compreensão e um

melhor acompanhamento dos algoritmos que vão sendo expostos, acrescido pela inclusão de exemplos de aplicação.

Nas duas subsecções que se seguem, são apresentados os algoritmos por ordem cronológica do seu aparecimento, enquadrados em duas categorias diferentes. Sempre que se faça referência a p e N , entendam-se como sendo o número de VTPs e o número de ligações na CCI, respectivamente.

3.3.1 Algoritmos de um passo

Este tipo de algoritmos caracteriza-se por gerar completamente, de uma vez só, o conjunto de VTPs a aplicar às ligações da CCI. São apresentados nove algoritmos desta categoria.

3.3.1.1 Algoritmo da pesquisa binária normal

Este algoritmo surge apresentado pela primeira vez em [Kau74], como um procedimento de teste de redes. É também descrito por outros autores, mas com nomes diferentes. Em [HRA88] encontra-se designado por "conjunto de vectores de teste de tamanho mínimo para detecção de curto-circuitos" e em [JY89] é designado por "algoritmo da sequência ascendente".

Em [Kau74] é provado que um conjunto de $\lceil \log_2 N \rceil$ VTPs é necessário e suficiente para detectar todos os possíveis curto-circuitos numa CCI com N ligações independentes. São gerados por este algoritmo $\lceil \log_2 N \rceil$ VTPs, segundo o procedimento da contagem binária ascendente. A cada ligação é aplicado um VTS com p bits, sendo aplicado o VTS $00 \dots 0$ (p 0s) à primeira ligação, recebendo as restantes ligações os VTSs seguintes da sequência binária ascendente.

Aplicando este algoritmo ao circuito da figura 3.1 resulta um conjunto com $\lceil \log_2 12 \rceil = 4$ VTPs como se mostra na tabela 3.1, onde se representam também as respostas esperadas e as síndromas existentes, supondo a ocorrência de curto-circuitos independentes entre as ligações $\langle 2, 7 \rangle$ e $\langle 4, 5 \rangle$.

Este algoritmo admite os VTSs $00 \dots 0$ (p 0s) e $11 \dots 1$ (p 1s), pelo que, nas ligações às quais estes são atribuídos, ligação 1 e ligação 2^p , as faltas do tipo fixo-a não são detectadas. Além disso, na ligação cujo VTS é $11 \dots 1$, as faltas do tipo circuito aberto permanecerão indetectadas (caso da tecnologia TTL) se a ligação tiver apenas uma entrada com o respectivo pino isolado do resto da ligação.

Ligação	VTS	VRS	Síndrome	
1	0000	0000	a -	
*	2	0001	0000	a c
	3	0010	0010	- -
*	4	0011	0000	a c
*	5	0100	0000	a c
	6	0101	0101	- -
*	7	0110	0000	a c
	8	0111	0111	- -
	9	1000	1000	- -
	10	1001	1001	- -
	11	1010	1010	- -
	12	1011	1011	- -

Tabela 3.1: Aplicação do algoritmo da pesquisa binária normal.

Relativamente às faltas do tipo curto-circuito, este algoritmo origina síndromas do tipo *aliasing* e do tipo *confounding*, como mostra a tabela 3.1. Estando as ligações (4,5) curto-circuitadas, resulta para estas um VRS comum coincidente com o VRS da ligação 1, a qual não está sob falta. Isto revela a incapacidade deste algoritmo para resolver conflitos deste tipo. Por outro lado, com as ligações (2,7) também curto-circuitadas, o VRS destas ligações é mais uma vez igual ao da ligação 1, mas dando agora origem a outro problema. De facto, o algoritmo detecta um curto-circuito entre as ligações (2, 4, 5, 7) que não existe, mostrando-se ineficaz na distinção de curto-circuitos independentes.

A complexidade temporal deste algoritmo é $O(N \log_2 N)$, em que o primeiro factor N reflecte a necessidade de cada VTP percorrer a cadeia de células BS. Apesar do número de células BS ser pelo menos o dobro de N , numa CCI completamente BST, mantém-se a ordem de complexidade indicada.

Como comentário final, este algoritmo apresenta óptimas características em relação à complexidade temporal, pois o número de VTPs gerados é mínimo, mas em contrapartida apresenta fraca capacidade de diagnóstico.

3.3.1.2 Algoritmo da pesquisa binária modificado

Encontra-se referido pela primeira vez em [GM82], como parte de um algoritmo adaptativo que será exposto na secção 3.3.2.1. Aparece também noutra literatura com a designação de "conjunto de vectores de teste de tamanho mínimo para detecção de curto-circuitos e faltas do tipo fixo-a" [HRA88] e "algoritmo da sequência ascendente modificado"

[JY89].

Este algoritmo baseia-se no algoritmo anterior, apresentando uma melhoria na detecção de faltas. Os VTSs $00 \dots 0$ e $11 \dots 1$ não são gerados por este algoritmo, continuando todos os outros VTSs a serem considerados, levando pois a um total de $\lceil \log_2(N + 2) \rceil$ VTPs. Deste modo, como nenhuma das N ligações possui atribuído o VTS $00 \dots 0$ ou $11 \dots 1$, as faltas do tipo circuito aberto, ou do tipo fixo-a, passam a ser detectadas.

Na tabela 3.2 mostram-se os vectores de teste obtidos por este algoritmo para o circuito modelo da figura 3.1, no qual se consideram curto-circuitos independentes entre as ligações $\langle 3, 5 \rangle$ e $\langle 7, 9 \rangle$, revelando o permanecimento das síndromas dos tipos *aliasing* e *confounding*.

Ligação	VTS	VRS	Síndrome	
1	0001	0001	a -	
2	0010	0010	- -	
*	3	0011	0001	a c
4	0100	0100	- -	
*	5	0101	0001	a c
6	0110	0110	- -	
★	7	0111	0001	a c
8	1000	1000	- -	
★	9	1001	0001	a c
10	1010	1010	- -	
11	1011	1011	- -	
12	1100	1100	- -	

Tabela 3.2: Aplicação do algoritmo da pesquisa binária modificado.

Relativamente à complexidade temporal continua a ser de $O(N \log_2 N)$. Enquanto esta característica é óptima, o mesmo não se pode dizer quanto à capacidade de diagnóstico, que pouco progrediu.

3.3.1.3 Algoritmo da pesquisa binária modificado com complemento

Apresentado pela primeira vez em [Wag87], é também referido em [HRA88] como "conjunto de vectores de teste para o diagnóstico de curto-circuitos e faltas do tipo fixo-a" e em [JY89] como "algoritmo de teste e diagnóstico com complemento".

Este algoritmo surge como uma evolução do algoritmo da pesquisa binária já apresentado na secção 3.3.1.1. Assim, são gerados $2 \lceil \log_2(N + 2) \rceil$ VTPs, com o objectivo de

aumentar a capacidade de diagnóstico resolvendo as ambiguidades causadas por síndromas do tipo *aliasing*. Cada um dos VTSs é composto por duas partes: os $p/2$ bits mais significativos (primeira metade) resultam da sequência de contagem binária ascendente, começando em $00 \dots 1$, e os $p/2$ bits menos significativos (segunda metade) são o complemento dos primeiros, surgindo daqui o nome do algoritmo.

Mostram-se na tabela 3.3 os resultados conseguidos por este algoritmo, considerando a existência de curto-circuitos independentes entre as ligações $\langle 1, 10 \rangle$ e $\langle 2, 9 \rangle$.

Ligação	VTS	VRS	Síndrome
* 1	00011110	00000100	- c
* 2	00101101	00000100	- c
3	00111100	00111100	- -
4	01001011	01001011	- -
5	01011010	01011010	- -
6	01101001	01101001	- -
7	01111000	01111000	- -
8	10000111	10000111	- -
* 9	10010110	00000100	- c
* 10	10100101	00000100	- c
11	10110100	10110100	- -
12	11000011	11000011	- -

Tabela 3.3: Aplicação do algoritmo da pesquisa binária modificado com complemento.

Tal como já acontecia no algoritmo anterior, todas as faltas do tipo fixo-a e do tipo circuito aberto são diagnosticáveis.

No que respeita a curto-circuitos, este algoritmo apresenta um melhoramento relativamente aos anteriores, porque nunca ocorrem situações do tipo *aliasing*. Para mostrar que assim é, apontam-se de seguida as condições que até aqui permitiam a ocorrência dum síndrome do tipo *aliasing* para cada posição dos bits dos VTSs em causa:

- em todos os VTSs é encontrado o mesmo valor (um 0 ou um 1);
- existe um 0 no VTS da ligação que não possui faltas e nos restantes VTSs existem os valores 0 e 1.

Note-se que a segunda condição é sempre necessária, senão o mesmo VTS seria atribuído a mais de uma ligação, sendo precisamente esta condição que garante a eliminação de

síndromas do tipo *aliasing* quando se aplica este algoritmo. Adicionando a cada VTS o seu próprio complemento, mantêm-se ambos os valores 0 e 1, mas o resultado dum curto-circuito do tipo e difere do 1 atribuído ao *bit* nessa posição do VTS da ligação sem faltas. Todos os aspectos aqui focados se podem verificar pela tabela 3.3.

O algoritmo da pesquisa binária modificado com complemento garante a não existência de síndromas do tipo *aliasing*, continuando porém a admitir síndromas do tipo *confounding*, conforme se mostram na tabela 3.3. Isto leva a que este algoritmo não garante o diagnóstico completo de faltas do tipo curto-circuito.

A complexidade temporal deste algoritmo continua a ser de $O(N \log_2 N)$. A sua capacidade de diagnóstico é melhor que a dos algoritmos anteriores, tendo sido obtida à custa do aumento do número de VTPs gerados, o dobro, de modo a só permanecerem por resolver as ambiguidades de diagnóstico do tipo *confounding*.

3.3.1.4 Algoritmo da sequência caminhante

A autoria deste algoritmo, como algoritmo de um só passo para teste de CCIs completamente BST, atribui-se a [HRA88]. No entanto, é já descrito em [GM82] como parte de um algoritmo adaptativo, e utilizado como um procedimento de teste de memórias [AR83]. Surge ainda referido como "algoritmo do 1 caminhante" em [JY89].

Ao contrário dos algoritmos anteriores, o algoritmo da sequência caminhante não se baseia no procedimento da pesquisa binária, através duma sequência de contagem binária ascendente. Este algoritmo gera N VTPs ($p \equiv N$), pelo que, cada VTS é constituído por N bits. À primeira ligação da CCI é atribuído um VTS com um 0 na posição mais à esquerda, sendo todos os restantes bits 1. A cada uma das ligações seguintes, é atribuído um VTS que é obtido a partir do VTS atribuído à ligação anterior, por deslocamento do 0 uma posição para a direita. Por fim, a N -ésima ligação terá um VTS caracterizado pela existência de um 0 na posição mais à direita e 1 nas restantes posições. A única razão para se escolher o 0 como *bit* a deslocar é ter-se considerado no modelo de faltas que um curto-circuito se comporta como um e lógico, em que 0 é o elemento absorvente. Se for considerado um comportamento do tipo ou, o *bit* 1 deverá ser usado como caminhante.

Na tabela 3.4 está ilustrado o conjunto de vectores gerados por este algoritmo e os respectivos vectores de resposta, considerando a existência de curto-circuitos independentes entre as ligações $\langle 3, 5 \rangle$ e $\langle 7, 8 \rangle$.

Todas as faltas do tipo fixo-a e circuito aberto são diagnosticáveis.

Ligação	VTS	VRS	Síndrome
0	011111111111	011111111111	--
1	101111111111	101111111111	--
2	110111111111	110111111111	--
*	3 111011111111	111010111111	--
4	111101111111	111101111111	--
*	5 111110111111	111010111111	--
6	111111011111	111111011111	--
*	7 111111101111	111111100111	--
*	8 111111110111	111111100111	--
9	111111111011	111111111011	--
10	111111111101	111111111101	--
11	111111111110	111111111110	--

Tabela 3.4: Aplicação do algoritmo da sequência caminhante.

As faltas do tipo curto-circuito são também completamente diagnosticáveis, pois com este algoritmo nunca ocorrem situações de *aliasing* nem de *confounding*, ao contrário do que acontecia nos algoritmos anteriores. A não ocorrência de quaisquer síndromas deve-se à propriedade de *independência diagonal* verificada pela matriz de vetores de teste [JY89]. No apêndice A, secção A.1, faz-se uma descrição desta propriedade, incluindo a respectiva demonstração da não ocorrência de quaisquer síndromas para um conjunto de vetores de teste que a verifique.

A não existência de síndromas é facilmente compreensível, atendendo a que sendo única a posição do 0 em cada VTS, nenhum curto-circuito gera um VRS com o primeiro 0 na mesma posição de outro VRS, correspondente a outro curto-circuito, ou igual a um VTS atribuído.

O diagnóstico completo de faltas permitido por este algoritmo é conseguido com um número de vetores de teste paralelos igual ao número de ligações da CCI. Se os VTPs fossem aplicados em paralelo, este algoritmo seria $O(N)$. Porém, num ambiente BST, a complexidade temporal resultante é $O(N^2)$, pois $p = N$.

3.3.1.5 Algoritmo do peso mínimo

Este algoritmo vem apresentado em [YJ89], exigindo a introdução dum novo conceito. Define-se *peso* dum vector como sendo o número de *bits* 0¹ presentes em cada VTS,

¹Será o número de 1s, se os curto-circuitos tiverem um comportamento do tipo ou.

representando-se por w .

A estratégia de geração de vectores de teste utilizada neste algoritmo situa-se entre a do algoritmo da pesquisa binária e a do algoritmo da sequência caminhante, podendo como primeiro passo escolher-se o número desejado de VTPs, de tal modo que sejam satisfeitas as restrições traduzidas pela inequação 3.1. Isto significa que p é limitado inferiormente pelo número de VTPs gerados pelo algoritmo da pesquisa binária modificado, e limitado superiormente pelo número de VTPs obtidos pelo algoritmo da sequência caminhante.

$$\lceil \log_2(N + 2) \rceil \leq p \leq N \quad (3.1)$$

À primeira ligação da CCI é aplicado um VTS com o *bit* mais à esquerda igual a 0 e os restantes $p - 1$ *bits* iguais a 1. Cada uma das ligações seguintes recebe um VTS obtido do anterior por deslocamento do 0 uma posição para a direita, sendo a posição libertada por este ocupada por um 1. Este procedimento é até aqui semelhante ao da sequência caminhante, sendo aplicado p vezes, resultando um conjunto de p VTSs com peso 1. Não se pretendendo o algoritmo da sequência caminhante, escolhe-se $p < N$. Assim, o próximo VTS, a atribuir à ligação $p + 1$, é gerado dum modo semelhante aos anteriores, mas atribuindo agora 0 às duas posições mais à esquerda e 1 às restantes posições. Para se obter o VTS seguinte, desloca-se o 0 mais à direita uma posição para a direita. Procede-se deste modo, até o referido 0 ocupar a posição do VTS mais à direita, após o que se passa a deslocar o outro 0, que até ali permanecia na posição mais à esquerda. Deste modo, obtém-se um conjunto de $p(p - 1)/2$ VTSs com peso 2, se ainda houver ligações com VTSs por atribuir. Este procedimento é repetido sucessivamente, originando VTSs com pesos crescentes até se obter o total de N VTSs.

O procedimento será melhor compreendido consultando a tabela 3.5, onde estão ilustrados, entre outros elementos, o conjunto de doze VTSs e respectivos pesos, obtidos por este algoritmo, considerando a geração de quatro VTPs² e a ocorrência de curto-circuitos independentes entre as ligações (3, 5) e (6, 8), do circuito que tem vindo a ser usado como exemplo.

Repare-se que para um dado peso w , o número de VTSs únicos N_w com p *bits* que se pode obter é dado pela expressão 3.2. Na figura 3.4 pode ver-se facilmente como evolui o referido número de VTSs.

²Note-se que para este exemplo podiam gerar-se até doze VTPs, caso em que seriam coincidentes com os obtidos pelo procedimento da sequência caminhante. Contudo, toma-se o valor mínimo de p .

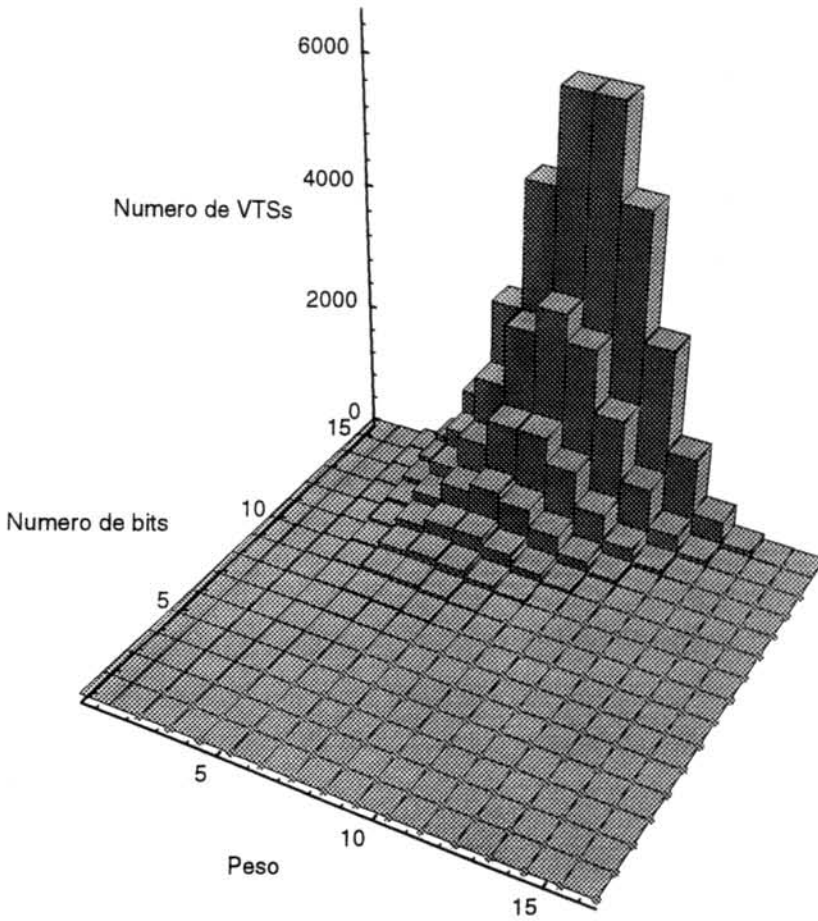


Figura 3.4: Quantidade de VTs possíveis em função do peso e do número de bits.

O peso máximo w_{max} dos N VTs é o menor valor de k que verifica a expressão 3.3.

$$N_w = \binom{p}{w} \quad (3.2)$$

$$\sum_{w=1}^k \binom{p}{w} \quad (3.3)$$

Utilizando-se $\lceil \log_2(N+2) \rceil$ VTPs, se todos os possíveis VTs (exceptuando $00 \dots 0$ e $11 \dots 1$) forem usados, caso em que o número de ligações N é tal que $\log_2(N+2) = \lceil \log_2(N+2) \rceil$, os VTs gerados por este algoritmo têm peso mínimo e são exactamente os mesmos que se obtêm pelo algoritmo da pesquisa binária modificado, embora atribuídos

Ligação	VTS	Peso	VRS	Síndrome	
1	0111	1	0111	- -	
2	1011	1	1011	- -	
*	3	1101	1	0001	a c
4	1110	1	1110	- -	
*	5	0011	2	0001	a c
★	6	0101	2	0001	a c
7	0110	2	0110	- -	
★	8	1001	2	0001	a c
9	1010	2	1010	- -	
10	1100	2	1100	- -	
11	0001	3	0001	a -	
12	0010	3	0010	- -	

Tabela 3.5: Aplicação do algoritmo do peso mínimo.

às ligações segundo uma ordem diferente. Nestas condições podem ocorrer síndromas dos tipos *aliasing* e *confounding*. Por outro lado, se forem utilizados N VTPs, melhora-se inquestionavelmente a capacidade de diagnóstico. De facto, considerando $p = N$, obtém-se exactamente o mesmo conjunto de VTSs que se obtém pelo algoritmo da sequência caminhante, tornando diagnosticáveis todas as faltas por não ocorrerem situações de *aliasing* ou *confounding*.

Pode assim dizer-se que a probabilidade de diagnosticar sem ambiguidades todas as faltas p_D , é directamente proporcional ao valor de p , variando este de acordo com a expressão 3.1. A expressão 3.4 mostra uma possível representação de proporcionalidade entre p_D e p , considerando o factor K_{nl} uma função não linear de p , com $K_{nl} = f(p)|_{p=N} = 1$.

$$p_D = K_{nl} \frac{p - \lceil \log_2(N + 2) \rceil}{N - \lceil \log_2(N + 2) \rceil} 100\% \quad (3.4)$$

Ocorrendo um curto-circuito do tipo e entre qualquer número de ligações, o VRS comum resultante terá mais *bits* 0 que qualquer dos VTSs correspondentes às ligações. Assim, a probabilidade de ocorrência de síndromas do tipo *aliasing* poderá ser minimizada optando por VTSs com o menor número possível de *bits* 0, ou seja, com VTSs de peso mínimo. A probabilidade de ocorrência de síndromas do tipo *confounding* decresce à medida que se aumenta o número de VTPs, pois quantos mais VTPs existirem menor é o número de VTSs que possuem o primeiro *bit* 0 na mesma posição. O exemplo que se mostra na tabela 3.5 revela que existem seis VTSs com o primeiro 0 na posição mais

à esquerda (ligações $\langle 1, 5, 6, 7, 11, 12 \rangle$), três VTSs com o primeiro 0 na segunda posição mais à esquerda (ligações $\langle 2, 8, 9 \rangle$), dois VTSs com o primeiro 0 na terceira posição mais à esquerda (ligações $\langle 3, 10 \rangle$) e um VTS com o primeiro 0 na posição mais à direita (ligação $\langle 4 \rangle$).

Pela análise feita conclui-se que a capacidade de diagnóstico deste algoritmo se situa entre as capacidades de diagnóstico do algoritmo da pesquisa binária modificado e do algoritmo da sequência caminhante. A complexidade temporal é $O(Np)$, situando-se p entre os limites estabelecidos pela expressão 3.1.

3.3.1.6 Algoritmo da independência máxima

Este algoritmo, tal como o anterior, encontra-se referido em [YJ89] pela primeira vez, necessitando de informação adicional para gerar os vectores de teste.

Caracterizado pela completa capacidade de diagnóstico, o algoritmo da independência máxima minimiza o número de VTPs a gerar, recorrendo a informação adicional sobre a CCI, que caracteriza a adjacência entre ligações e a extensão máxima para as faltas do tipo curto-circuito, isto é, a indicação do número máximo de ligações σ que poderão estar curto-circuitadas entre si. Antes de se apresentar o algoritmo, expõem-se a seguir alguns conceitos [YJ89] necessários à sua formulação.

Dado um VTS em que i e j correspondam à primeira e última posições nas quais exista um 0, define-se o *peso potencial* \tilde{w} deste VTS como sendo o valor dado pela expressão 3.5. Para o VTS $00 \dots 0$, define-se $\tilde{w} = 0$.

$$\tilde{w} = j - i + 1 \quad (3.5)$$

Decorre desta definição que, o número de VTSs únicos com p bits $N_{\tilde{w}}$, a que corresponde um peso potencial \tilde{w} , é dado pela expressão 3.6.

$$N_{\tilde{w}} = \begin{cases} 1 & \Leftarrow \tilde{w} = 0 \\ p & \Leftarrow \tilde{w} = 1 \\ (p - \tilde{w} + 1)2^{\tilde{w}-2} & \Leftarrow \tilde{w} > 1 \end{cases} \quad (3.6)$$

Nesta expressão, $p - \tilde{w} + 1$ representa o número de VTSs que se podem obter começando com o 0 da posição i na posição mais à esquerda e deslocando para a direita o VTS completo até o 0 da posição j atingir a posição mais à direita. O factor $2^{\tilde{w}-2}$ representa o número de VTSs que podem ser gerados de cada um dos prévios $p - \tilde{w} + 1$ VTSs, considerando todas as combinações diferentes dos *bits* entre as posições i e j .

Diz-se que um conjunto com N ligações $\{n_1, n_1, \dots, n_N\}$ está *ordenado por adjacência*, se n_i for mais adjacente, isto é, mais provável que venha a estar em curto-circuito, com n_{i+1} do que com n_{i+2} , para $1 \leq i \leq N - 2$, e se n_i for mais adjacente com n_{i-1} do que com n_{i-2} , para $3 \leq i \leq N$. Note-se que a tarefa de ordenar por adjacência as ligações duma CCI, pode não ser de fácil resolução, pelo que apenas será praticável uma ordenação aproximada. Por exemplo, numa CCI, uma ligação pode ser fisicamente adjacente a outra ligação em mais de uma dimensão. Uma ordenação aproximada pode ser obtida tomando a adjacência entre os pinos dos CIs pertencentes a cada ligação, assumindo que os curto-circuitos mais comuns são causados pela soldadura, afectando os pinos de CIs fisicamente adjacentes.

Considere-se um conjunto de ligações ordenado por adjacência, no qual as ligações n_i e n_j ($1 \leq i, j \leq N$) representam as ligações de menor e maior ordem, respectivamente, afectadas pelo mesmo curto-circuito. Define-se *extensão* e desse curto-circuito como sendo o valor dado pela expressão 3.7.

$$e = j - i + 1 \quad (3.7)$$

Do ponto de vista físico, o valor de e está relacionado com o valor de σ , o número de ligações afectadas por um curto-circuito, estando também relacionado com a ordenação das ligações: $e = \sigma$ para uma ordenação completa por adjacência e $e \geq \sigma$ para uma ordenação por adjacência parcial. Repare-se que em qualquer das situações $2 \leq e \leq N$. O maior valor que e pode tomar representa-se por E , obedece à desigualdade dupla $2 \leq e \leq N$, sendo um limite pré-determinado.

Descreve-se de seguida o procedimento de geração dos vectores de teste decomposto em cinco fases:

1. Determine-se o menor valor de p de modo a permitir diagnosticar sem ambiguidades todos os curto-circuitos esperados, supondo um limite E para o número máximo de ligações envolvidas num curto-circuito, de acordo com a expressão 3.8. Esta expressão encontra-se deduzida no apêndice A, secção A.2.

$$p = \begin{cases} [E + \log_2(N + 1) - \log_2 E - 1] & \Leftrightarrow E > 2 \vee \log_2(N + 1) < [\log_2(N + 1)] \\ [\log_2(N + 2)] & \Leftrightarrow E = 2 \wedge \log_2(N + 1) = [\log_2(N + 1)] \end{cases} \quad (3.8)$$

2. Forme-se uma lista das N ligações da CCI, ordenadas por adjacência, utilizando adjacência parcial se a adjacência completa for impraticável.
3. Construam-se conjuntos de VTSs com p bits, excluindo os VTSs $00 \dots 0$ e $11 \dots 1$, tais que, cada subconjunto seja constituído por todos os VTSs com o mesmo peso potencial e o mesmo peso de Hamming [Bla83], isto é, o mesmo número de elementos não nulos. Note-se que cada subconjunto contém o número máximo de VTSs únicos que são diagonalmente independentes, surgindo daqui o nome de algoritmo da independência máxima.
4. Ordene-se o subconjunto de VTSs com o peso potencial 1 e acrescente-se repetidamente o próximo subconjunto de VTSs com peso potencial mínimo, até que existam N VTSs no conjunto assim formado. Se dois ou mais subconjuntos de VTSs tiverem o mesmo peso potencial, use-se em primeiro lugar aquele que tiver menor peso de Hamming.
5. Atribua-se, sequencialmente, um VTS do conjunto obtido no passo anterior à próxima ligação da lista ordenada por adjacência, até que todas as N ligações tenham um VTS único.

A tabela 3.6 mostra a aplicação deste algoritmo ao circuito da figura 3.1, considerando $E = 4$ e as ligações ordenadas por adjacência. Nela se representam os VTSs, os respectivos pesos potenciais, as respostas obtidas e as síndromas, supondo que as ligações $\langle 2, 3 \rangle$ estão curto-circuitadas. Para melhor compreender a sequência de geração de VTSs, sugere-se a consulta da segunda coluna da referida tabela.

A principal propriedade deste algoritmo é que, desde que a extensão e dum curto-circuito não exceda o limite superior E que foi admitido, e desde que não existam faltas múltiplas numa mesma ligação, existirá sempre uma capacidade de diagnóstico completa. Isto deve-se ao conjunto de ligações estar ordenado por adjacência e de quaisquer E VTSs consecutivos possuírem a propriedade da independência diagonal (veja-se o apêndice A, secção A.1). No exemplo da tabela 3.6, apesar dos VRSs das ligações em curto-circuito coincidirem com o VRS da ligação 7, não existe um síndrome do tipo *aliasing*, porque o facto do conjunto de ligações estar ordenado por adjacência impossibilita esta última ligação de estar envolvida no curto-circuito, sem que as ligações $\langle 4, 5, 6 \rangle$ também o estejam.

A complexidade temporal deste algoritmo é $O(Np)$, com p escolhido de modo a garantir o total diagnóstico de faltas, para uma extensão máxima de curto-circuitos predeterminada. No entanto, se todas as ligações puderem estar curto-circuitadas, ou seja, $E \equiv N$,

Ligação	VTS	Peso potencial	VRS	Síndroma	
1	01111	1	01111	--	
*	2	10111	1	10011	--
*	3	11011	1	10011	--
4	11101	1	11101	--	
5	11110	1	11110	--	
6	00111	2	00111	--	
7	10011	2	10011	--	
8	11001	2	11001	--	
9	11100	2	11100	--	
10	01011	3	01011	--	
11	10101	3	10101	--	
12	11010	3	11010	--	

Tabela 3.6: Aplicação do algoritmo da independência máxima.

a complexidade temporal resultante é $O(N^2)$. Recorde-se que p depende do número de ligações N da CCI e da extensão máxima dos curto-circuitos E , conforme mostra a expressão 3.8.

Como comentário final, observe-se que este algoritmo apresenta um excelente equilíbrio entre o número de VTPs gerados e a capacidade de diagnóstico, que é requerida ser completa. A razão das suas boas características deve-se à informação adicional sobre a CCI a testar.

3.3.1.7 Algoritmo do auto-diagnóstico

Este algoritmo vem apresentado em [CLW90], tendo como característica principal o facto de gerar um conjunto de vectores de teste que garante que qualquer ligação da CCI envolvida num curto-circuito apresentará um VRS diferente do seu VTS. O conjunto de vectores gerado pelo algoritmo da pesquisa binária modificado com complemento [Wag87], já apresentado anteriormente, também possui esta característica, embora o presente algoritmo permita chegar à mesma capacidade de diagnóstico gerando um menor número de VTPs.

Como qualquer ligação apresentará um VRS diferente do respectivo VTS, sob o efeito duma falta, a identificação de ligações envolvidas em faltas faz-se precisamente pela análise dos VRSs originados, dando-se a esta propriedade o nome de *auto-diagnóstico*.

Este algoritmo é apresentado com base no conceito de *cobertura* entre vectores [CLW90], dizendo-se que um VTS v_1 cobre um VTS v_2 , se e só se v_1 tem um 0 em todas as posições

em que v_2 tem um 0^3 . Está aqui a supor-se um comportamento do tipo *e* para os curto-circuitos. Por exemplo, o VTS 00010 cobre o VTS 00011. Um conjunto de VTSs diz-se *independente*, se nenhum dos VTSs cobrir outro VTS. Repare-se que um conjunto de VTSs que disponha desta propriedade garantirá a inexistência de síndromas do tipo *aliasing*, uma vez que o VRS produzido por um conjunto de ligações em curto-circuito só poderia coincidir com o VTS atribuído a uma outra ligação, se o VTS desta cobrisse os VTSs das ligações envolvidas no curto-circuito.

Segundo este algoritmo, o procedimento de geração de vectores de teste gera todos os VTSs de p bits, com $\lfloor p/2 \rfloor$ bits 1. Deste modo, o número de VTPs gerados será o valor dado pela expressão 3.9.

$$p = \min \left\{ p_i : \binom{p_i}{\lfloor p_i/2 \rfloor} \geq N \right\} \quad (3.9)$$

O primeiro VTS contém os $\lfloor p/2 \rfloor$ bits 1 nas posições mais à esquerda. O segundo VTS obtém-se do primeiro, por deslocação do 1 mais à direita, uma posição para a direita. Esta deslocação é praticada para obter os VTSs seguintes, até o 1 atingir a posição mais à direita. Nessa altura, o VTS seguinte obtém-se do primeiro VTS, por deslocação dos dois bits 1 mais à direita, uma posição para a direita, e para o VTS seguinte desloca-se, se ainda possível, o bit 1 mais à direita. Este procedimento é repetido até serem gerados os N VTSs necessários. Pela tabela 3.7 compreende-se facilmente o método de geração dos VTSs.

Continuando com o circuito da figura 3.1 para fins de exemplificação, a aplicação deste algoritmo gera seis VTPs, pois $N = 12$. Ilustram-se na tabela 3.7 os VTSs, VRSs e síndromas existentes, obtidos pelo algoritmo do auto-diagnóstico, admitindo curto-circuitos independentes entre as ligações $\langle 3, 5 \rangle$ e $\langle 6, 9 \rangle$.

Apesar de garantir a inexistência de ambiguidades do tipo *aliasing*, este algoritmo não impede a ocorrência de situações do tipo *confounding*, como se pode constatar pela tabela 3.7. Possuindo a mesma capacidade de diagnóstico que o algoritmo da pesquisa binária modificado com complemento, este algoritmo apresenta no entanto a vantagem de gerar um menor número de VTPs, para o mesmo número de ligações. Quanto à sua complexidade temporal é $O(Np)$, com p a tomar o valor dado pela expressão 3.9.

³Note-se que v_1 também pode ter um 0 em posições em que v_2 possui um 1.

Ligação	VTS	VRS	Síndrome	
1	111000	111000	- -	
2	110100	110100	- -	
*	3	110010	100000	- c
4	110001	110001	- -	
*	5	101100	100000	- c
*	6	101010	100000	- c
7	101001	101001	- -	
8	100110	100110	- -	
*	9	100101	100000	- c
10	100011	100011	- -	
11	011100	011100	- -	
12	011010	011010	- -	

Tabela 3.7: Aplicação do algoritmo do auto-diagnóstico.

3.3.1.8 Algoritmo do diagnóstico global

O algoritmo do diagnóstico global [CLW90] garante a inexistência de síndromas do tipo *aliasing*, embora não garanta que qualquer ligação envolvida num curto-circuito apresente um VRS diferente do respectivo VTS, ao contrário do que acontece com o algoritmo anterior. Apesar de conservar a mesma capacidade de diagnóstico característica dos algoritmos do auto-diagnóstico e da pesquisa binária modificado com complemento, este algoritmo é o que, de entre os três, gera o menor número de vectores de teste. Esta redução torna-se possível pelo facto de os dois referidos algoritmos conterem condições suficientes, mas não necessárias, para garantir a inexistência de síndromas na forma de *aliasing*.

A exposição deste algoritmo é feita com base no conceito de *esparsidade* de um conjunto de vectores de teste [CLW90], que se pode definir como sendo a propriedade pela qual um conjunto de ligações envolvidas num mesmo curto-circuito apresentará um VRS que será diferente de qualquer VTS atribuído a uma ligação não incluída no curto-circuito.

O procedimento de geração dos vectores de teste do algoritmo do diagnóstico global é definido recursivamente. Por conveniência de formulação, considere-se o conjunto $T(i, 1)$, definido como o conjunto de todos os VTSs com i bits gerados pelo algoritmo do auto-diagnóstico, e o conjunto $T(i, 2)$, definido como o conjunto de todos os VTSs com i bits gerados por este algoritmo do diagnóstico global. Nestas circunstâncias, $T(p+1, 2)$ consiste na reunião de dois subconjuntos disjuntos, $X(p+1)$ e $Y(p+1)$, em que $X(p+1)$ contém todos os VTSs de $T(p, 2)$, com um 1 acrescentado à direita, e $Y(p+1)$ contém todos os

VTSs de $T(p, 1)$, com um 0 acrescentado à direita. Os conjuntos iniciais são $T(1, 1) = \{1\}$ e $T(1, 2) = \{1, 0\}$. Uma vez que o tamanho de $T(p+1, 2)$ é dado pela soma do tamanho de $T(p, 2)$ com $\binom{p}{\lfloor p/2 \rfloor}$, resulta que $p+1$ vectores de teste serão suficientes para testar $2 + \sum_{k=1}^p \binom{k}{\lfloor k/2 \rfloor}$ ligações. Para uma CCI com N ligações, o número de VTPs gerados por este algoritmo é traduzido pela expressão 3.10.

$$p = \min\{p_i : 2 + \sum_{k=1}^{p_i-1} \binom{k}{\lfloor k/2 \rfloor} \geq N\} \quad (3.10)$$

Na tabela 3.8 apresentam-se os resultados da aplicação deste algoritmo, considerando curto-circuitos independentes entre as ligações $\langle 1, 2, 5 \rangle$, $\langle 4, 9 \rangle$ e $\langle 7, 11 \rangle$.

Ligação	VTS	VRS	Síndroma
* 1	11111	01011	- -
* 2	01111	01011	- -
3	10111	10111	- -
* 4	10011	10000	- c
* 5	01011	01011	- -
6	11001	11001	- -
◇ 7	10101	10000	- c
8	01101	01101	- -
* 9	11000	10000	- c
10	10100	10100	- -
◇ 11	10010	10000	- c
12	01010	01010	- -

Tabela 3.8: Aplicação do algoritmo do diagnóstico global.

Apesar de este algoritmo garantir a inexistência de síndromas do tipo *aliasing*, pode tornar-se necessária uma análise mais complexa do conjunto de VRSs resultantes. Com efeito, torna-se necessário verificar se o VRS produzido por um conjunto de ligações em curto-circuito, coincide com algum dos VTSs que lhes foram atribuídos. Só desta forma se torna possível concluir se nenhuma das ligações que apresentam VRSs correctos estará envolvida num curto-circuito; tal o ilustra o caso das ligações $\langle 1, 2, 5 \rangle$, que se apresenta na tabela 3.8. É com base no conceito de esparsidade de um conjunto de VTSs que se faz tal análise, pois existindo um conjunto de ligações com um VRS comum igual ao VTS

de uma das ligações, esta está forçosamente envolvida no curto-circuito que se averigua. Tal como no algoritmo anterior, este algoritmo admite a existência de síndromas do tipo *confounding*, conforme se ilustra no exemplo apresentado na tabela 3.8.

Este algoritmo consegue atingir a mesma capacidade de diagnóstico do algoritmo do auto-diagnóstico com um menor número de VTPs, necessitando porém duma análise mais detalhada dos VRSs. Relativamente à complexidade temporal é $O(Np)$, sendo p o valor fornecido pela expressão 3.10.

3.3.1.9 Algoritmo do diagnóstico máximo

Este algoritmo, cuja autoria se deve a [LB91], destaca-se dos anteriores por permitir que qualquer ligação dum CCI possa estar sob o efeito simultâneo de faltas múltiplas. Em situações deste tipo, algumas das faltas podem não ser detectadas pelos algoritmos apresentados anteriormente, pois conforme se disse no início da sua apresentação, iria supor-se que cada ligação só poderia estar sob o efeito de uma falta única.

Para a formulação do algoritmo são assumidas as falhas físicas do tipo circuito aberto e curto-circuito. Numa ligação em que existam várias falhas do tipo circuito aberto, estas são modeladas como se fossem uma simples falta do tipo circuito aberto. Do mesmo modo, se uma ligação estiver curto-circuitada com outra em vários pontos da ligação física, assume-se um curto-circuito simples entre elas. Numa mesma ligação podem ocorrer simultaneamente faltas do tipo circuito aberto e curto-circuito. Numa situação destas, o valor lógico captado nas células BST associadas a pinos de entrada, isto é, pontos de observação, é determinado pelo efeito combinado de ambos os tipos de faltas, tendo em consideração o modelo de faltas adoptado.

Em [LB91] estende-se o conceito de cobertura entre vectores para cobertura de um conjunto de vectores, apresentando-se ainda novos conceitos. Seja V_A o resultado de um curto-circuito do tipo e envolvendo as ligações cujos VTSs formam o conjunto de vectores $\{V_{a_1}, V_{a_2}, \dots, V_{a_k}\}$. Um vector V_i cobre o conjunto de vectores $\{V_{a_1}, V_{a_2}, \dots, V_{a_k}\}$ se V_i cobre V_A .

Uma *síndrome de cobertura dum conjunto* é uma síndrome que resulta dum curto-circuito envolvendo um conjunto de ligações W' que cobre um VRS ou é coberto pelo VRS dum ligação n_i não incluída em W' .

Seja S um conjunto de vectores de teste e $W = \{n_1, n_2, \dots, n_N\}$ um conjunto de ligações. O conjunto S diz-se com *independência de cobertura* se nenhum elemento de S

cobre, nem é coberto, pela intersecção de quaisquer subconjuntos de S , ou seja, se não existe síndrome de cobertura do conjunto para nenhum VTS de S .

Uma falta f diz-se não diagnosticável se não existir um conjunto de vectores de teste S e um algoritmo A , tal que, pela aplicação de S à CCI e análise das respostas usando A , f possa ser identificada. Note-se que todas as faltas simples são diagnosticáveis, não se podendo dizer o mesmo das faltas múltiplas. A figura 3.5 ilustra exemplos de faltas não diagnosticáveis.

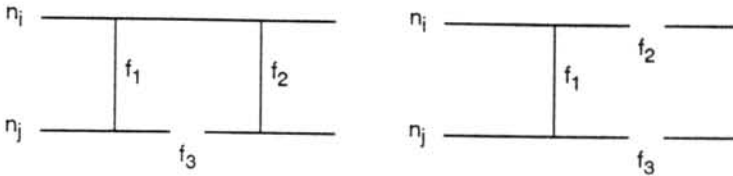


Figura 3.5: Faltas não diagnosticáveis.

Os vários níveis de capacidade de diagnóstico possíveis de atingir por diferentes algoritmos, são caracterizados em [LB91] e referidos como *resolução do diagnóstico*, fazendo-se uma análise dos principais algoritmos, já aqui apresentados, quanto a esta característica.

Nas figuras 3.6, 3.7 e 3.8, apresentam-se situações de faltas múltiplas, que os algoritmos anteriores não conseguem detectar, mesmo verificando a independência diagonal e a independência do conjunto de VTSs. Nestes exemplos, considera-se excepcionalmente que os curto-circuitos têm um comportamento lógico do tipo **ou**.

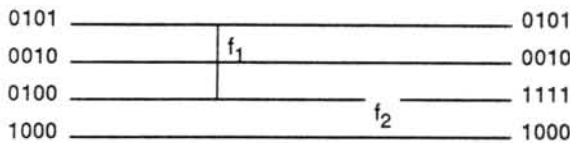


Figura 3.6: Curto-circuito não identificado por uma sequência diagonalmente independente.

Na figura 3.6 a falta f_1 não é identificada pelo conjunto de vectores representado. Na figura 3.7 a falta f_2 também não é identificada. Apresentam-se de seguida as razões pelas quais estas faltas, f_1 e f_2 respectivamente, não são identificadas.

- Um conjunto de vectores de teste S não permite identificar um curto-circuito entre duas ligações n_i e n_j , se existir um circuito aberto mais próximo do ponto de ob-

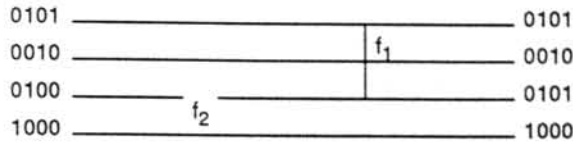


Figura 3.7: Circuito aberto não identificado por uma sequência diagonalmente independente.

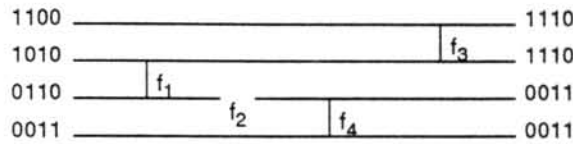


Figura 3.8: Curto-circuito não identificado por um conjunto de vetores independente.

servação da ligação n_i do que o curto-circuito, e se o VTS de n_i é coberto pelo VTS de n_j .

- Um conjunto de vetores de teste S não permite identificar um circuito aberto numa ligação n_i , se existir outra ligação n_j tal que um curto-circuito entre n_i e n_j se localiza mais próximo do ponto de observação da ligação n_i do que o circuito aberto, e se o VTS de n_i é coberto pelo VTS de n_j .

Na figura 3.8, a falta f_1 , do tipo curto-circuito, não é identificada por um conjunto de VTSs independente, porque 0110 é coberto pelo conjunto de VTSs que origina a resposta 1110, tornando impossível determinar se a falta f_1 existe ou não.

Define-se de seguida o termo *diagnóstico máximo*. Seja $W = \{n_1, n_2, \dots, n_N\}$ um conjunto de ligações, D_i o conjunto de pontos de controlo (*drivers*) da ligação n_i e R_i o conjunto de pontos de observação (*receivers*) da mesma ligação. Um conjunto de VTSs S pode atingir o diagnóstico máximo para W quando se verificarem as duas condições que se seguem.

1. Para todas as ligações n_i da CCI, analisando as respostas resultantes da aplicação do conjunto de vetores S , pode ser determinada a existência de continuidade entre D_i e R_i .

A ligação entre D_i e R_i existe, se para todos os valores de k , cada ponto de controlo $d_{ik} \in D_i$ consegue transferir o seu valor lógico correctamente a todos os pontos de

observação pertencentes a R_i . Diz-se que um ponto de controlo de D_i transfere o seu valor lógico a um ponto de observação de R_i se o VRS de n_i cobre o respectivo VTS. Note-se que para cada ligação só um ponto de controlo pode estar activo num dado instante. Por outras palavras, se a ligação entre D_i e R_i existe, então a aplicação do VTS de n_i ao respectivo ponto de controlo activo torna verdadeira uma das afirmações: o VRS de n_i é igual ao respectivo VTS, ou o VRS de n_i cobre o respectivo VTS.

2. Para todas as ligações n_i e n_j , $i \neq j$, analisando as respostas resultantes da aplicação do conjunto de vectores S , pode ser determinada a existência de continuidade entre D_i e R_j .

A ligação entre D_i e R_j não existe se para todos os valores de k , cada ponto de controlo $d_{ik} \in D_i$ não consegue transferir o seu valor lógico a qualquer ponto de observação pertencente a R_j . Por outras palavras, se a ligação entre D_i e R_j não existe, então a aplicação do VTS de n_i ao respectivo ponto de controlo activo torna verdadeira uma das afirmações: o VRS de n_j é diferente do VTS de n_i , ou o VRS de n_j não cobre o VTS de n_i .

É enunciado e demonstrado em [LB91] um teorema referindo que um algoritmo é capaz de identificar todas as ligações com faltas e capaz de identificar todas as faltas que são diagnosticáveis se e só se o conjunto de vectores de teste gerado pelo algoritmo atingir o diagnóstico máximo. Na sequência deste teorema é enunciado e demonstrado um outro que diz que um conjunto de vectores de teste pode atingir o diagnóstico máximo para um conjunto de ligações se e só se esse conjunto de vectores possuir independência de cobertura, bastando para tal que o conjunto de vectores de teste possua uma subsequência de vectores gerados pelo procedimento da sequência caminhante.

Após estas considerações, que tiveram por objectivos elucidar o leitor das deficiências existentes nos algoritmos já apresentados quando no modelo de faltas se incluem faltas múltiplas, e expôr novos conceitos e características dum conjunto de vectores de teste, passa-se à apresentação do algoritmo do diagnóstico máximo.

Considere-se $W = \{n_1, n_2, \dots, n_N\}$ um conjunto de ligações a testar e seja $N_{br}(n_i) \subset W$ o conjunto de ligações vizinhas de n_i . O procedimento que a seguir se indica gera um conjunto de vectores atingindo o diagnóstico máximo, resultando daqui o nome de algoritmo do diagnóstico máximo.

1. Construa-se um grafo de vizinhanças $NG = (V, E)$, tal que:

- (a) Para cada $n_i \in W$ existe o correspondente nó $v_i \in V$.
- (b) O conjunto de ramos é $E = E_1 \cup E_2$, onde

$$E_1 = \{e : e = (v_i, v_j, \forall n_j \in N_{br}(n_i), \forall n_i \in W)\}$$

$$E_2 = \{e : e = (v_j, v_k, \forall v_j, v_k \in N_{br}(n_i), \forall v_i \in V)\}$$

Os conjuntos E_1 e E_2 incluem, respectivamente, os ramos entre cada ligação e cada uma das suas ligações vizinhas, e os ramos entre ligações que são vizinhas de uma mesma ligação.

2. Atribua-se a cada nó $v_i \in V$ uma cor $C(v_i)$, tal que $C(v_i) \neq C(v_j)$ se $e = (v_i, v_j) \in E$, e que o número de cores c seja mínimo.
3. Associe-se a cada cor um VTS único com c bits, de modo a que em cada VTS apenas um bit tome o valor lógico 0⁴.
4. Atribua-se a cada ligação $n_i \in W$ o VTS associado à cor que foi atribuída ao respectivo nó $v_i \in V$.

A figura 3.9 ilustra o grafo resultante da aplicação deste método ao circuito da figura 3.1, considerando as vizinhanças de ligações, $N_{br}(n_1) = \{n_2\}$, $N_{br}(n_2) = \{n_1, n_3\}$, $N_{br}(n_3) = \{n_2, n_4\}$, $N_{br}(n_4) = \{n_3, n_5\}$, $N_{br}(n_5) = \{n_4, n_6\}$, $N_{br}(n_6) = \{n_5, n_7\}$, $N_{br}(n_7) = \{n_6, n_8\}$, $N_{br}(n_8) = \{n_7, n_9\}$, $N_{br}(n_9) = \{n_8, n_{10}\}$, $N_{br}(n_{10}) = \{n_9, n_{11}\}$, $N_{br}(n_{11}) = \{n_{10}, n_{12}\}$, $N_{br}(n_{12}) = \{n_{11}\}$, e a existência de curto-circuitos, restritos a grupos de ligações vizinhas, entre as ligações $\langle 3, 4 \rangle$, $\langle 5, 6 \rangle$ e $\langle 11, 12 \rangle$. Conforme se vê pelo grafo, três cores V, A e B, são necessárias e suficientes, atribuindo-lhes os VTSs 011, 101 e 110, respectivamente, segundo o procedimento apresentado. Na tabela 3.9 indicam-se os VTSs e os VRSs resultantes da consideração da figura 3.9.

O conjunto de vectores é composto apenas por três vectores diferentes, donde resulta a possibilidade de ocorrerem quatro VRSs distintos (001, 010, 100, 000) para os curto-circuitos possíveis, considerando as vizinhanças existentes entre as ligações. Repare-se que o facto de as ligações $\langle 5, 6, 11, 12 \rangle$ apresentarem o mesmo VRS, não corresponde porém a uma situação de ambiguidade, pois o algoritmo reconhece que as ligações $\langle 5, 6 \rangle$ não são vizinhas das ligações $\langle 11, 12 \rangle$, tornando estes dois curto-circuitos independentes distinguíveis.

⁴1 se o curto-circuito tiver um comportamento lógico do tipo ou.

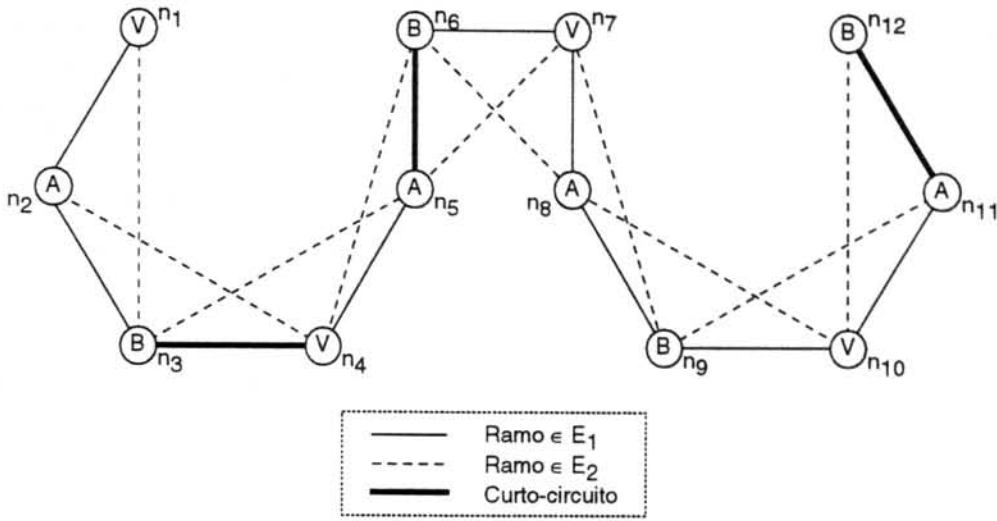


Figura 3.9: Grafo de vizinhanças e atribuição de cores.

Ligação	VTS	VRS	Síndrome	
1	011	011	--	
2	101	101	--	
*	3	110	010	--
*	4	011	010	--
*	5	101	100	--
*	6	110	100	--
7	011	011	--	
8	101	101	--	
9	110	110	--	
10	011	011	--	
◇	11	101	100	--
◇	12	110	100	--

Tabela 3.9: Aplicação do algoritmo do diagnóstico máximo.

A completa capacidade de detecção e diagnóstico com um conjunto de vectores de tamanho mínimo caracteriza este algoritmo. Porém, possui a significativa desvantagem de necessitar de uma lista de vizinhanças para cada ligação, dada a dificuldade de estabelecer um conceito de vizinhança e à extracção da relação de vizinhanças para uma dada CCI. É ainda de referir que o passo 2 requer a resolução de um problema do tipo NP-completo [AHU74].

Encontra-se exposto em [MM93] um algoritmo que supõe um modelo de faltas baseado na adjacência dos pinos, cujo autor refere como sendo um novo algoritmo. Porém, analisando este algoritmo, referido como "algoritmo de diagnóstico baseado na adjacência dos pinos", rapidamente se reconhece nele o algoritmo do diagnóstico máximo publicado dois anos antes. Refira-se ainda que em [MM93] não é feita qualquer referência ao trabalho de [LB91], estranhando-se o desconhecimento deste último. O algoritmo é descrito para o modelo de faltas já referido como uma heurística, não recorrendo a qualquer formalismo matemático.

Após a descrição do modelo de faltas, são apresentadas algumas definições. Para que o leitor possa reconhecer nelas o equivalente exposto em [LB91], apresentam-se de seguida as mais significativas. Para uma dada ligação em particular, designada por *ligação primária*, o *conjunto de ligações primárias de curto-circuito* é formado pelas ligações com possibilidades de constituírem um curto-circuito com a ligação primária considerada, sendo o *conjunto de ligações secundárias de curto-circuito* formado pelas ligações primárias de curto-circuito das ligações primárias de curto-circuito da ligação em causa. Estes dois conjuntos de ligações identificam-se precisamente com os conjuntos de ramos E_1 e E_2 do grafo VG utilizado no algoritmo do diagnóstico máximo.

3.3.2 Algoritmos adaptativos

Os algoritmos adaptativos distinguem-se dos algoritmos de um passo pelo facto de gerarem numa primeira fase um conjunto mínimo de VTPs, usado para efeitos de detecção de faltas, e depois então gerarem um segundo conjunto de VTPs baseado nos resultados obtidos, usado para efeitos de diagnóstico de faltas. Este segundo grupo de algoritmos que se apresenta é composto por seis algoritmos adaptativos.

3.3.2.1 Algoritmo de teste dos W vectores

O algoritmo que aqui se descreve foi apresentado pela primeira vez em [GM82], onde o autor descreve uma metodologia de teste designada por ECIPT (*Electronic Chip-In-Place Test*). Contudo, a designação escolhida foi retirada de [JY89], onde o algoritmo se

encontra resumido.

O primeiro passo deste algoritmo consiste na geração do mesmo conjunto de VTSs com $\lceil \log_2(N + 2) \rceil$ bits do algoritmo da pesquisa binária modificado. Tal como foi descrito na altura, a existência de eventuais síndromas dos tipos *aliasing* e *confounding* não permite que a capacidade de diagnóstico seja completa, tornando este primeiro passo do algoritmo útil apenas do ponto de vista de detecção de faltas. Para resolver as ambiguidades do diagnóstico existentes é gerado um segundo grupo de VTSs, cujo tamanho é igual ao número de ligações W cujos VRSs diferem da resposta esperada. O procedimento de geração deste segundo conjunto de W VTSs coincide com o do algoritmo da sequência caminhante, atribuindo-se a cada ligação defeituosa um VTS com o elemento caminhante 0 a deslocar-se sucessivamente da esquerda para a direita, enquanto às ligações isentas de faltas se atribui o VTS $11 \dots 1$. Caso exista um grande número de faltas $W \rightarrow N$, pelo que, no limite, esta segunda fase do algoritmo se reduz ao algoritmo da sequência caminhante. Contudo, na prática W assume valores muito inferiores a N .

Apresentam-se na tabela 3.10 os dois conjuntos de VTSs gerados por este algoritmo adaptativo, assim como os respectivos VRSs e síndromas existentes, assumindo curto-circuitos independentes entre as ligações $\langle 3, 5 \rangle$ e $\langle 7, 9 \rangle$ do circuito habitual. Como o número de ligações com defeito considerado é $W = 4$, resulta que o segundo conjunto de vectores gerados é composto por quatro VTPs.

Ligação	Primeiro passo			Segundo passo		
	VTS	VRS	Síndroma	VTS	VRS	Síndroma
1	0001	0001	a -	1111	1111	--
2	0010	0010	--	1111	1111	--
* 3	0011	0001	a c	0111	0011	--
4	0100	0100	--	1111	1111	--
* 5	0101	0001	a c	1011	0011	--
6	0110	0110	--	1111	1111	--
* 7	0111	0001	a c	1101	1100	--
8	1000	1000	--	1111	1111	--
* 9	1001	0001	a c	1110	1100	--
10	1010	1010	--	1111	1111	--
11	1011	1011	--	1111	1111	--
12	1100	1100	--	1111	1111	--

Tabela 3.10: Aplicação do algoritmo de teste dos W vectores.



O segundo passo deste algoritmo garante a inexistência de síndromas, porque os W vectores por ele gerados satisfazem a propriedade da independência diagonal (apêndice A, secção A.1). Deste modo, resulta uma capacidade de diagnóstico completa. No entanto, a existência de faltas múltiplas pode levar a que ligações com defeitos proporcionem VRSs correctos, pelo que os resultados obtidos nestas circunstâncias serão insatisfatórios. Será apresentada uma situação destas quando for apresentado, na secção 3.3.2.5, o algoritmo adaptativo A1.

Para este algoritmo, o número total de VTPs é $p = \lceil \log_2(N + 2) \rceil + W$, sendo a sua complexidade temporal função do número de ligações defeituosas. Para uma CCI sem faltas, a complexidade temporal resultante coincide com a do algoritmo da pesquisa binária modificado, sendo $O(N \log_2 N)$. Na situação mais desfavorável, em que todas as N ligações possuem faltas, a complexidade temporal resultante coincide com a do algoritmo da sequência caminhante que é $O(N^2)$. No entanto, atendendo a que o número de faltas é habitualmente bastante inferior ao número de ligações da CCI, a complexidade temporal efectiva deste algoritmo é bastante menor que a do algoritmo da sequência caminhante.

3.3.2.2 Algoritmo do vector único

O algoritmo do vector único, apresentado pela primeira vez em [JY89], é o algoritmo adaptativo dual do algoritmo de um passo da pesquisa binária modificado com complemento, no sentido em que ambos apresentam a mesma capacidade de diagnóstico.

O primeiro passo consiste em gerar os mesmos $\lceil \log_2(N + 2) \rceil$ VTPs do algoritmo da pesquisa binária modificado, tal como no algoritmo anterior, com o propósito de detectar as faltas existentes. Pela análise das síndromas verifica-se a existência de *aliasing* e *confounding*. Se as síndromas são do tipo *aliasing*, e não do tipo *confounding*, então é necessário um segundo passo do teste, para resolver as situações de *aliasing*. Este segundo passo consiste em gerar mais um VTP adicional. Assim, às ligações não defeituosas cujo VRS é igual ao VRS de ligações com faltas, é atribuído um 0, enquanto às restantes ligações é atribuído um 1. Deste modo, todas as síndromas do tipo *aliasing* são verificadas em paralelo, com um único vector adicional.

Na tabela 3.11 apresentam-se os dois conjuntos de VTSs gerados por este algoritmo adaptativo, assim como os respectivos VRSs e síndromas existentes, assumindo circuitos independentes entre as ligações $\langle 3, 5 \rangle$, $\langle 7, 9 \rangle$ e $\langle 11, 12 \rangle$ do circuito habitual.

Este algoritmo não garante o diagnóstico completo de faltas, uma vez que o VTP do segundo passo apenas permite esclarecer e eliminar as situações de *aliasing*, permanecendo

Ligação	Primeiro passo			Segundo passo			
	VTS	VRS	Síndroma	VTS	VRS	Síndroma	
1	0001	0001	a -	0	0	- -	
2	0010	0010	- -	1	1	- -	
*	3	0011	0001	a c	1	1	- c
4	0100	0100	- -	1	1	- -	
*	5	0101	0001	a c	1	1	- c
6	0110	0110	- -	1	1	- -	
*	7	0111	0001	a c	1	1	- c
8	1000	1000	a -	0	0	- -	
*	9	1001	0001	a c	1	1	- c
10	1010	1010	- -	1	1	- -	
◇	11	1011	1000	a -	1	1	- -
◇	12	1100	1000	a -	1	1	- -

Tabela 3.11: Aplicação do algoritmo do vector único.

as síndromas do tipo *confounding* resultantes do primeiro passo. Por este motivo se atribui a este algoritmo adaptativo a dualidade com o algoritmo da pesquisa binária modificado com complemento [Wag87], assistindo-se a uma solução de compromisso entre o número de VTPs e a complexidade da análise das respostas, para uma mesma capacidade de diagnóstico.

O número total de VTPs gerados por este algoritmo é $p = \lceil \log_2(N+2) \rceil + 1$, sendo a sua complexidade temporal $O(N \log_2 N)$. Apesar de ser o algoritmo adaptativo que produz um menor número de VTPs, a sua capacidade de diagnóstico é severamente afectada pelo facto de o segundo passo não considerar a eliminação de síndromas do tipo *confounding*. Esta desvantagem torna-se significativa, uma vez que o segundo conjunto de vectores de um algoritmo adaptativo tem por objectivo proporcionar o diagnóstico completo, aproveitando toda a informação conseguida pelo primeiro conjunto de vectores.

3.3.2.3 Algoritmo dos C vectores óptimos

Apresentado pela primeira vez em [JY89], este algoritmo adaptativo apresenta a mesma capacidade de diagnóstico do algoritmo da sequência caminhante. No segundo passo deste algoritmo são analisadas as síndromas existentes com o objectivo de decidir o que fazer, ao contrário do que acontece com o algoritmo de teste dos W vectores, onde é gerada de imediato uma sequência caminhante para as ligações que apresentam faltas.

Com o objectivo de detectar as faltas existentes, e eventuais síndromas, este algoritmo gera o primeiro conjunto de VTSs de acordo com o algoritmo da pesquisa binária modificado, conforme acontece nos dois algoritmos adaptativos que antecedem este. O segundo conjunto de vectores será gerado se existirem síndromas dos tipos *aliasing* ou *confounding*, de acordo com o procedimento que se segue, onde é usado o conceito de grau de *confounding* introduzido na secção 3.1.

1. Se existirem síndromas apenas do tipo *aliasing*, então para se obter uma completa capacidade de diagnóstico, basta gerar um vector adicional, de acordo com o algoritmo do vector único.
2. As restantes síndromas podem ser do tipo *confounding* ou simultaneamente *aliasing* e *confounding*. As situações de *confounding* resolvem-se do seguinte modo: seja $C = \max\{c_i\}$ o maior grau de *confounding* das síndromas em causa. Um total de $C - 1$ VTPs é necessário para esclarecer as situações de *confounding*, sendo a estratégia adoptada para gerar estes vectores idêntica à do algoritmo da sequência caminhante, e explicada mais à frente.
3. Por fim, se ainda existirem síndromas do tipo *aliasing*, é necessário gerar mais um vector de teste, de acordo com o procedimento já descrito para esclarecer estas situações. Assim sendo, se as síndromas forem dos tipos *aliasing* e *confounding*, então $C - 1$ vectores são suficientes para resolver as situações de *confounding* e um vector adicional é necessário para resolver as situações de *aliasing*. Em geral, no máximo C VTPs são suficientes para diagnosticar completamente todos os curto-circuitos em que as ligações estejam envolvidas.

No passo 2 deste procedimento faz-se referência aos $C - 1$ VTPs suficientes para resolver os problemas de *confounding*. Descreve-se de seguida o modo de os gerar. O primeiro VTP é gerado atribuindo um 0 a uma das ligações envolvidas na primeira falta do número máximo de faltas consideradas independentes, sendo isto feito para todas as síndromas do tipo *confounding* existentes, e às restantes ligações é atribuído um 1. O segundo VTP é gerado atribuindo um 0 a uma das ligações envolvidas na segunda falta do número máximo de faltas consideradas independentes, e mais uma vez, isto é feito para todas as faltas do tipo *confounding*, e às restantes ligações é atribuído um 1. Este processo é repetido até o $(C - 1)$ -ésimo, e último vector, ser gerado. Como C é o maior grau de *confounding* presente, este processo de solucionar as síndromas do tipo *confounding* termina primeiro para as síndromas de menor grau. Quando isto acontece, a todas as ligações envolvidas nestas síndromas é atribuído um 1 nos VTPs seguintes. Quando o

$(C - 1)$ -ésimo VRP é capturado, o conjunto de VRPs obtido deste modo é suficiente para concluir se a última falta considerada independente é realmente independente ou não, sendo deste modo necessários $C - 1$ VTPs apenas. O número máximo de VTPs gerados por este algoritmo é dado por $p = \lceil \log_2(N + 2) \rceil + C$.

Como exemplo de aplicação deste algoritmo, apresentam-se na tabela 3.12 os dois conjuntos de VTSs gerados, assim como os respectivos VRSs e síndromas existentes, assumindo a existência de curto-circuitos independentes entre as ligações $\langle 3, 5 \rangle$, $\langle 6, 9 \rangle$ e $\langle 7, 8 \rangle$.

Ligação	Primeiro passo			Segundo passo		
	VTS	VRS	Síndroma	VTS	VRS	Síndroma
1	0001	0001	a -	10	10	- -
2	0010	0010	- -	11	11	- -
* 3	0011	0001	a -	11	11	- -
4	0100	0100	- -	11	11	- -
* 5	0101	0001	a -	11	11	- -
* 6	0110	0000	- c	01	01	- -
◇ 7	0111	0000	- c	11	11	- -
◇ 8	1000	0000	- c	11	11	- -
* 9	1001	0000	- c	11	01	- -
10	1010	1010	- -	11	11	- -
11	1011	1011	- -	11	11	- -
12	1100	1100	- -	11	11	- -

Tabela 3.12: Aplicação do algoritmo dos C vectores óptimos.

Conforme se pode constatar, as ligações 3 e 5 estão numa situação de *aliasing* com a ligação 1, a qual não se encontra envolvida em qualquer curto-circuito. A única situação de *confounding* existente envolve as ligações 6, 7, 8 e 9. Como existem ambos os tipos de síndromas, e o maior grau de *confounding* existente é $C = 2$, pois só existe uma situação de *confounding*, o segundo passo do algoritmo começa por considerar um VTP, o qual permite clarificar a situação de *confounding*, mostrando a existência de dois curto-circuitos independentes. No entanto, como a situação de *aliasing* continua por resolver, é necessário mais um VTP, ficando assim completo, com dois VTPs, o segundo passo do algoritmo.

Com este algoritmo adaptativo é possível atingir um diagnóstico completo de faltas. Como o número de VTPs é inferior ao correspondente do algoritmo de teste dos W vectores, este algoritmo dos C vectores óptimos apresenta uma boa solução de compromisso

entre o número de VTPs gerados e a ambiguidade do diagnóstico, à custa duma complexidade adicional. De facto, a tarefa de identificar o número máximo de curto-circuitos independentes possíveis em cada síndrome do tipo *confounding* é complexa, devido ao grande número de combinações possíveis, mesmo quando o número de ligações envolvidas numa situação de *confounding* não é elevado. Portanto, apesar de garantir o esclarecimento de todas as síndromas dos tipos *aliasing* e *confounding*, com o máximo de $\lceil \log_2(N+2) \rceil + C$ vectores de teste, a geração do segundo conjunto de vectores de teste implica o procedimento mais complexo dos que foram apresentados até agora.

Relativamente à complexidade temporal deste algoritmo, ela depende não só do número de ligações com VRSs que revelam faltas, mas também das características das síndromas presentes. Mais uma vez, se o circuito não possuir faltas, a complexidade temporal será $O(N \log_2 N)$, correspondendo à do algoritmo da pesquisa binária modificado. Repare-se ainda que, mesmo que todas as ligações estejam envolvidas em faltas, o tamanho do segundo conjunto de vectores será menor que aquele que seria gerado pelo algoritmo da sequência caminhante, contrariamente ao que acontece no algoritmo de teste dos W vectores. Considerando que o número e características dos curto-circuitos que normalmente ocorrem são tais que o número de VTPs gerados pelo segundo passo do algoritmo não é muito significativo, a complexidade temporal efectiva deste algoritmo adaptativo será muito semelhante a $O(N \log_2 N)$.

3.3.2.4 Algoritmo do diagnóstico em dois passos

Este algoritmo encontra-se apresentado pela primeira vez em [CLW90], apresentando a particularidade de gerar ambos os conjuntos de vectores de teste de início, sendo o segundo conjunto independente das respostas obtidas para o primeiro conjunto, tornando a designação de algoritmo adaptativo algo imprópria.

O primeiro conjunto de vectores é determinado pela aplicação a cada ligação da CCI de um VTS com $\lceil \log_2 N \rceil$ bits, gerados por uma sequência de contagem binária ascendente, com início no VTS $00 \dots 0$, tal como no primeiro algoritmo [Kau74] apresentado neste capítulo. Se nenhuma falta é detectada com este primeiro conjunto de VTSs, então é desnecessário o segundo conjunto de VTSs para efeitos de diagnóstico. O segundo conjunto, quando necessário, é constituído por $\lceil \log_2(\lceil \log_2 N \rceil + 1) \rceil$ VTPs, representando cada VTS o número de bits 0, no correspondente VTS do primeiro conjunto.

Na tabela 3.13 apresentam-se os dois conjuntos de VTSs gerados por este algoritmo, assim como os respectivos VRSs e síndromas existentes, considerando a ocorrência de curto-circuitos independentes entre as ligações $\langle 2, 9 \rangle$ e $\langle 3, 5 \rangle$.

Ligação	Primeiro passo			Segundo passo			
	VTS	VRS	Síndroma	VTS	VRS	Síndroma	
1	0000	0000	a -	100	100	- -	
*	2	0001	0000	a c	011	011	- c
*	3	0010	0000	a c	011	011	- c
	4	0011	0011	- -	010	010	- -
*	5	0100	0000	a c	011	011	- c
	6	0101	0101	- -	010	010	- -
	7	0110	0110	- -	010	010	- -
	8	0111	0111	- -	001	001	- -
*	9	1000	0000	a c	011	011	- c
	10	1001	1001	- -	010	010	- -
	11	1010	1010	- -	010	010	- -
	12	1011	1011	- -	001	001	- -

Tabela 3.13: Aplicação do algoritmo do diagnóstico em dois passos.

Foi referido a propósito do algoritmo do auto-diagnóstico que um conjunto de VTSs independente não dá origem a síndromas do tipo *aliasing*. No caso concreto deste algoritmo, considerando dois VTSs quaisquer, $u = u_1 u_2$ e $v = v_1 v_2$, em que u_1 e v_1 , e u_2 e v_2 são, respectivamente, as componentes dos vectores u e v gerados no primeiro e no segundo passo do algoritmo, se u_1 cobre v_1 , então u_2 é um número superior a v_2 , pelo que, u_2 não cobre v_2 . Daqui resulta que nenhum vector cobre outro vector, tornando a reunião dos dois conjuntos de vectores independente. Quanto a síndromas do tipo *confounding*, podem continuar a existir. No exemplo apresentado na tabela 3.13 nota-se a incapacidade de o segundo conjunto de vectores resolver a situação de *confounding*, pois nem sequer detectou os curto-circuitos existentes. Contudo, o *aliasing* existente foi eliminado.

O número total de VTPs gerados por este algoritmo é dado pela expressão 3.11.

$$p = \lceil \log_2 N \rceil + \lceil \log_2(\lceil \log_2 N \rceil + 1) \rceil \quad (3.11)$$

A capacidade de diagnóstico é igual à do algoritmo da pesquisa binária modificado com complemento [Wag87], pelo que se podem designar por similares [CLW90]. O algoritmo da pesquisa binária modificado com complemento, também pode ser encarado como adaptativo, uma vez que a geração da segunda metade do conjunto de VTPs poderá ser feita apenas quando se pretender diagnosticar faltas. A complexidade do algoritmo do diagnóstico em dois passos é $O(N \log_2 N)$.

3.3.2.5 Algoritmo adaptativo A1

O primeiro algoritmo adaptativo exposto em [LB91] vem referido pelo autor como algoritmo adaptativo A1. Tal como o algoritmo do diagnóstico máximo já descrito, também este algoritmo adopta um modelo de faltas em que é admitida a existência de faltas múltiplas numa mesma ligação.

Apresenta-se de seguida o procedimento de geração dos vectores de teste organizado em três fases.

1. O primeiro conjunto de vectores a aplicar é obtido do mesmo modo que foi já anteriormente descrito para o algoritmo do auto-diagnóstico, possuindo a característica de ser um conjunto independente, ou seja, um conjunto em que nenhum dos VTSs cobre qualquer outro VTS. O número de VTPs assim gerados é dado pela expressão 3.12.

$$p_1 = \min \left\{ p_i : \binom{p_i}{\lfloor p_i/2 \rfloor} \geq N \right\} \quad (3.12)$$

Se não for detectada nenhuma falta, então o procedimento termina aqui.

2. O conjunto de ligações é dividido em dois grupos, colocando-se no grupo 0 todas as ligações que apresentem um VRS único e igual ao correspondente VTS, sendo as restantes ligações incluídas no grupo 1.
3. O segundo conjunto de vectores, correspondente ao segundo passo do algoritmo adaptativo, obtém-se pela atribuição a cada ligação do grupo 1 de um VTS com o elemento caminhante 0 a deslocar-se sucessivamente da esquerda para a direita, e às restantes ligações, pertencentes ao grupo 0, atribui-se o VTS $11 \dots 1$.

Na tabela 3.14 apresentam-se os dois conjuntos de vectores gerados por este algoritmo para o exemplo que tem vindo a ser utilizado, assim como as respectivas respostas e síndromas existentes, supondo a existência de curto-circuitos independentes entre as ligações $\langle 3, 5 \rangle$, $\langle 6, 9 \rangle$ e $\langle 11, 12 \rangle$.

Comparando este algoritmo com o algoritmo de teste dos W vectores, conclui-se que diferem apenas no procedimento utilizado para gerar o primeiro conjunto de vectores, sendo este procedimento mais ambicioso a ponto de não dar origem a síndromas do tipo *aliasing*. Assim, a sequência caminhante parcial utilizada para gerar o segundo conjunto de vectores, tem apenas por objectivo eliminar síndromas do tipo *confounding*. Note-se que as ligações 11 e 12 também pertencem ao grupo 1, embora não tenham originado

Ligação	Primeiro passo			Segundo passo			
	VTS	VRS	Síndroma	VTS	VRS	Síndroma	
1	111000	111000	--	111111	111111	--	
2	110100	110100	--	111111	111111	--	
*	3	110010	100000	-c	011111	001111	--
4	110001	110001	--	111111	111111	--	
*	5	101100	100000	-c	101111	001111	--
*	6	101010	100000	-c	110111	110011	--
7	101001	101001	--	111111	111111	--	
8	100110	100110	--	111111	111111	--	
*	9	100101	100000	-c	111011	110011	--
10	100011	100011	--	111111	111111	--	
◇	11	011100	011000	--	111101	111100	--
◇	12	011010	011000	--	111110	111100	--

Tabela 3.14: Aplicação do algoritmo adaptativo A1.

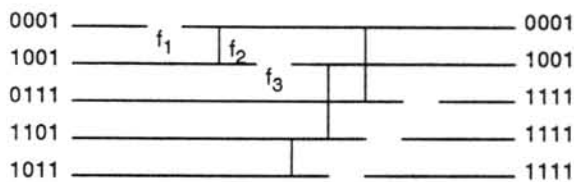


Figura 3.10: Situação com faltas múltiplas.

uma situação de *confounding*, e por isso também lhes são atribuídos VTSs originados pela sequência caminhante.

A propósito do algoritmo de teste dos W vectores, este poderá levar a resultados insatisfatórios se for considerada a existência de faltas múltiplas numa mesma ligação, conforme se ilustra no exemplo apresentado na figura 3.10. Repare-se que o facto de as duas primeiras ligações ilustradas produzirem VRSs únicos, e iguais aos respectivos VTSs, conduziria a que as faltas f_1 , f_2 e f_3 , não fossem detectadas. Contudo, empregando um primeiro conjunto de VTSs independente, como no caso do algoritmo adaptativo A1, mesmo na presença de faltas múltiplas é mantida uma completa capacidade de detecção e diagnóstico.

O número total de VTPs gerados por este algoritmo é dado pela expressão 3.13, onde p_1 é dado pela expressão 3.12, e W é o número de faltas detectadas no primeiro passo do algoritmo, isto é, o número de ligações do grupo 1.

$$p = p_1 + W \quad (3.13)$$

Quanto à complexidade temporal, irá depender do valor de W . Os valores extremos são $O(Np_1)$ e $O(N^2)$, ocorrendo respectivamente, numa situação em que no primeiro passo não são detectadas faltas, e numa situação em que todas as ligações estão afectadas. Os limites de complexidade temporal variam pois entre os valores que assumem para o algoritmo do auto-diagnóstico e para o algoritmo da sequência caminhante.

3.3.2.6 Algoritmo adaptativo A2

O segundo algoritmo adaptativo exposto em [LB91] vem referido pelo autor como algoritmo adaptativo A2. Tal como o algoritmo anterior, também este usa um modelo de faltas que permite a existência de faltas múltiplas numa mesma ligação.

Apresenta-se de seguida o procedimento de geração dos vectores de teste organizado em cinco fases.

1. O primeiro conjunto de vectores a aplicar é obtido do mesmo modo que foi já anteriormente descrito para o algoritmo do auto-diagnóstico. O número de VTPs gerados é dado pelo valor mínimo de p_1 que satisfaz a expressão 3.14.

$$p_1 = \min\left\{p_i : \begin{pmatrix} p_i \\ \lfloor p_i/2 \rfloor \end{pmatrix} \geq N\right\} \quad (3.14)$$

Se não for detectada nenhuma falta, então o procedimento termina aqui.

2. O conjunto de ligações é dividido em dois grupos, colocando-se no grupo 0 todas as ligações que apresentem um VRS único e igual ao correspondente VTS, sendo as restantes ligações incluídas no grupo 1.
3. O grupo 1 de ligações é dividido em subgrupos, de modo a agrupar as ligações que possuem o mesmo VRS. Numerem-se estes subgrupos de 1 a G , e designe-se por K a cardinalidade do maior destes subgrupos.
4. A primeira parte do segundo conjunto de vectores é obtida pela atribuição de um VTS com o elemento caminhante 0, a deslocar-se da esquerda para a direita, a cada ligação presente nos G subgrupos constituídos no passo anterior, enquanto às restantes ligações é atribuído um 1. Deste modo, são gerados K VTPs, uma vez que este processo decorre em paralelo. Se $K = 1$, então não é necessário gerar nenhum VTP nesta fase.

5. A segunda parte do segundo conjunto de vectores é obtida pela atribuição sucessiva de um VTS com o elemento caminhante 0, a deslocar-se da esquerda para a direita, a cada um dos G subgrupos constituídos no passo 3, sendo atribuído o mesmo VTS a todas as ligações pertencentes ao mesmo subgrupo. Às ligações que constituem o grupo 0 é atribuído um 1. Deste modo, são gerados G VTPs neste passo.

Na tabela 3.15 apresentam-se os dois conjuntos de vectores gerados por este algoritmo para o exemplo que tem vindo sistematicamente a ser utilizado, assim como as respectivas respostas e síndromas existentes, supondo a ocorrência de curto-circuitos independentes entre as ligações $\langle 3, 5 \rangle$, $\langle 6, 9 \rangle$ e $\langle 11, 12 \rangle$. Repare-se através da tabela, que a primeira parte do segundo conjunto de VTSs está representada no lado direito da coluna referente a estes VTSs.

Ligação	Primeiro passo			Segundo passo					
	VTS	VRS	Síndroma	VTS		VRS		Síndroma	
1	111000	111000	--	11	1111	11	1111	--	
2	110100	110100	--	11	1111	11	1111	--	
*	3	110010	100000	-c	01	0111	01	0011	--
4	110001	110001	--	11	1111	11	1111	--	
*	5	101100	100000	-c	01	1011	01	0011	--
*	6	101010	100000	-c	01	1101	01	1100	--
7	101001	101001	--	11	1111	11	1111	--	
8	100110	100110	--	11	1111	11	1111	--	
*	9	100101	100000	-c	01	1110	01	1100	--
10	100011	100011	--	11	1111	11	1111	--	
◇	11	011100	011000	--	10	0111	10	0011	--
◇	12	011010	011000	--	10	1011	10	0011	--

Tabela 3.15: Aplicação do algoritmo adaptativo A2.

Neste exemplo da tabela 3.15, foram escolhidas as mesmas situações de curto-circuito já usadas no exemplo de aplicação do algoritmo anterior, o que conduziu ao mesmo número total de vectores gerados. Contudo, em geral, o número de vectores gerados pelo algoritmo adaptativo A2 é inferior ao gerado pelo algoritmo adaptativo A1. Como exemplo disso, suponham-se três situações diferentes de *confounding*, envolvendo cada uma delas quatro ligações. Então, este algoritmo iria gerar sete VTPs e o anterior daria origem a doze VTPs.

$$p = p_1 + G + K \quad (3.15)$$

O número total de VTPs gerados por este algoritmo é dado pela expressão 3.15, onde p_1 é o valor resultante da expressão 3.14, G é o número de subgrupos constituídos pelas ligações que apresentam o mesmo VRS e K é a dimensão do maior destes subgrupos. A complexidade temporal deste algoritmo depende não só de G como também depende de K . Os valores extremos são $O(Np_1)$ e $O(N^2)$, tal como no algoritmo anterior. A situação $O(N^2)$ ocorre quando, no pior dos casos, for $G = 1$ e $K = N$. Mas conforme já foi dito, em termos de complexidade temporal, este algoritmo tem um melhor comportamento que o anterior.

3.4 Comentários finais

Reunem-se no capítulo seguinte importantes conclusões acerca do que neste capítulo foi tratado e exposto.

Capítulo 4

Análise comparativa dos algoritmos

A diversidade de algoritmos apresentados implica a necessidade de uma reflexão relativamente à aplicabilidade de cada um e ao significado das suas principais diferenças. A aplicabilidade de um algoritmo é essencialmente condicionada pelas dificuldades de implementação que apresenta, que são especialmente relevantes para os algoritmos que requerem informações adicionais, tais como as relações de vizinhança entre ligações.

As secções que compõem este capítulo reúnem os aspectos mais marcantes que caracterizam os algoritmos apresentados no capítulo anterior, sintetizados na tabela 4.1, com o objectivo de facilitar uma comparação em termos globais. Faz-se ainda a escolha de um algoritmo que englobe as melhores características.

4.1 Número de vectores de teste

A segunda coluna da tabela 4.1 apresenta as expressões que revelam o número de VTPs gerados por cada algoritmo. Da sua análise conclui-se que possuem um comportamento logarítmico e/ou linear em função do número de ligações.

A figura 4.1 ilustra seis funções que mostram a evolução dos p VTPs gerados por algoritmos de um só passo com o número de ligações da CCI. As duas primeiras correspondem ao algoritmo da pesquisa binária e pesquisa binária modificado, respectivamente. A função seguinte, figura 4.1 c), identifica-se manifestamente com o algoritmo da sequência caminhante, pela linearidade entre p e W apresentada. No algoritmo da independência máxima, o número de VTPs depende não só do número de ligações N como também do número de ligações que podem estar curto-circuitadas entre si E . É importante notar que

Algoritmo	Número de VTPs gerados	Capacidade de diagnóstico	Complexidade temporal
Pesquisa binária normal	$\lceil \log_2 N \rceil$	Fraca: possível ocorrência de <i>aliasing</i> e <i>confounding</i>	$O(N \log_2 N)$
Pesquisa binária modificado	$\lceil \log_2(N+2) \rceil$	Fraca: possível ocorrência de <i>aliasing</i> e <i>confounding</i>	$O(N \log_2 N)$
Pesquisa binária modificado com complemento	$2\lceil \log_2(N+2) \rceil$	Não ocorre <i>aliasing</i> mas pode ocorrer <i>confounding</i>	$O(N \log_2 N)$
Sequência caminhante	N	Completa	$O(N^2)$
Peso mínimo	$\lceil \log_2(N+2) \rceil \leq p \leq N$	Quanto maior o valor de p escolhido menor será a ambiguidade existente	$O(Np)$
Independência Máxima	$p = f(N, E)$	Completa	$O(Np)$
Auto-diagnóstico	$\min\{p_i : \binom{p_i}{\lfloor p_i/2 \rfloor} \geq N\}$	Não ocorre <i>aliasing</i> mas pode ocorrer <i>confounding</i>	$O(Np)$
Diagnóstico Global	$\min\{p_i : 2 + \sum_{k=1}^{p_i-1} \binom{k}{\lfloor k/2 \rfloor} \geq N\}$	Não ocorre <i>aliasing</i> mas pode ocorrer <i>confounding</i>	$O(Np)$
Diagnóstico Máximo	$p = f(c)$	Completa	$O(Np)$
Teste dos W vectores	$\lceil \log_2(N+2) \rceil + W$	Completa	$O(N \log_2 N) \leq O \leq O(N^2)$
Vector único	$\lceil \log_2(N+2) \rceil + 1$	Situações de <i>confounding</i> podem ficar por resolver	$O(N \log_2 N)$
C vectores óptimos	$\lceil \log_2(N+2) \rceil + C$	Completa	$O(N \log_2 N)$
Diagnóstico em dois passos	$\lceil \log_2 N \rceil + \lceil \log_2(\lceil \log_2 N \rceil + 1) \rceil$	Situações de <i>confounding</i> podem ficar por resolver	$O(N \log_2 N)$
Adaptativo A1	$\min\{p_i : \binom{p_i}{\lfloor p_i/2 \rfloor} \geq N\} + W$	Completa	$O(Np_1) \leq O \leq O(N^2)$
Adaptativo A2	$\min\{p_i : \binom{p_i}{\lfloor p_i/2 \rfloor} \geq N\} + G + K$	Completa	$O(Np_1) \leq O \leq O(N^2)$

Tabela 4.1: Características quantitativas e qualitativas dos algoritmos apresentados.

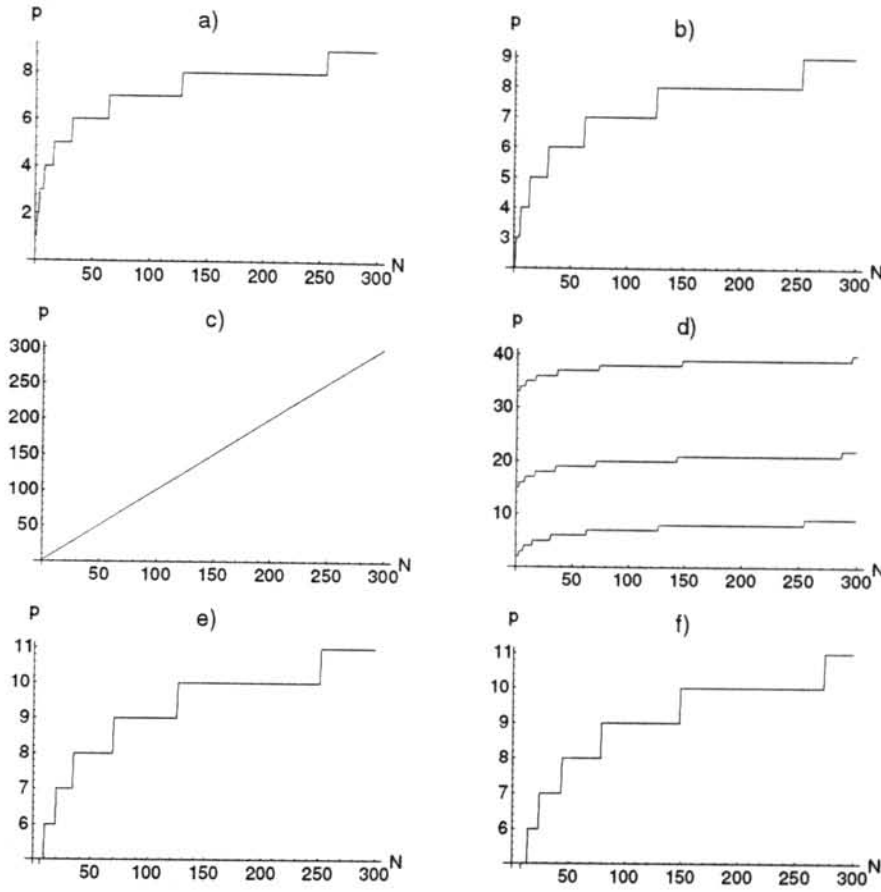


Figura 4.1: As funções $p = f(N)$ em algoritmos de um só passo.

este algoritmo é o único que apresenta uma completa capacidade de diagnóstico recorrendo a um só passo, e que o número de vectores não é exageradamente elevado, dependendo isso da topologia da carta através do parâmetro E . A figura 4.1 d) mostra esta dependência para três valores de E : 2, 18 e 36. Finalmente, nas figuras 4.1 e) e figura 4.1 f) ilustram-se as expressões de p para os algoritmos do auto-diagnóstico e diagnóstico global, respectivamente.

A figura 4.2 contém ilustradas funções que se identificam com $p = f(N)$ para os algoritmos adaptativos. A primeira delas corresponde ao algoritmo de teste dos W vectores, onde se consideram os valores 0, 36 e 75 para W . Na figura 4.2 b) ilustra-se a mesma

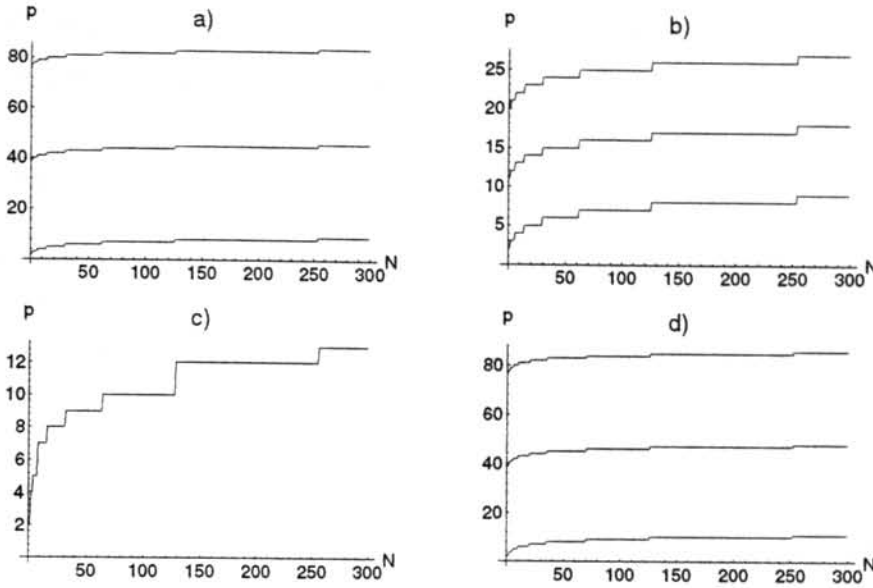


Figura 4.2: As funções $p = f(N)$ em algoritmos adaptativos.

evolução para o algoritmo dos C vectores óptimos para valores de C iguais a 0, 10 e 18. Desde já se realça este algoritmo por manifestar uma capacidade de diagnóstico completa recorrendo a um reduzido número de VTPs. Por fim, a figura 4.2 c) representa p para o algoritmo do diagnóstico em dois passos, e a figura 4.2 d) corresponde ao algoritmo adaptativo A1, considerando os mesmos valores para o número de faltas detectadas W que foram utilizadas na figura 4.2 a).

4.2 Capacidade de diagnóstico

Com a excepção do algoritmo da sequência caminhante e da independência máxima, nenhum dos restantes algoritmos de um só passo permite o diagnóstico completo de faltas. Esta é uma importante limitação, porque a reparação de falhas em CCIs necessita que o diagnóstico seja o mais completo possível [FMdJ89]. No entanto, o tempo requerido para a aplicação dos N VTPs no algoritmo caminhante é proibitivamente longo [FMdJ89] e a informação acerca da CCI exigida pelo algoritmo da independência máxima pode nem sempre ser fácil de disponibilizar.

O teste de produção apresenta requisitos diferentes ao nível do procedimento de teste e diagnóstico, quando comparados com a reparação de CCIs. A existência deste diferentes requisitos sugere os algoritmos adaptativos como uma atraente escolha. Quando é requi-

rido apenas a detecção de faltas, é usado o conjunto de VTPs de tamanho mínimo gerados pelo primeiro passo. Por outro lado, a combinação do primeiro e segundo passos garante que seja atingível o completo diagnóstico de faltas.

4.3 Conclusão

A escolha dum algoritmo para efectuar diagnóstico de faltas não pode ser feita em termos absolutos, isto é, não existe um algoritmo que seja o melhor entre todos para qualquer situação de diagnóstico. Conforme o nível de diagnóstico exigido têm de ponderar-se as consequências originadas por aspectos inerentes ao próprio método de operação do algoritmo, tais como a dimensão do conjunto de VTPs necessários e a complexidade quer em obtê-los quer em analisar as síndromas resultantes. Contudo, num cenário de teste que exija uma boa resolução do diagnóstico de faltas, os algoritmos adaptativos revelam-se como os mais interessantes e capazes, e entre estes, o algoritmo dos C vectores óptimos é aquele que atinge o diagnóstico completo com o mínimo de vectores de teste.

Capítulo 5

Implementação dos algoritmos adaptativos

Mostrou-se no capítulo 3 o modo como os vários algoritmos geram vectores de teste, como detectam faltas, se existirem, e como as diagnosticam. Em alguns dos algoritmos concluiu-se que era necessária uma análise complexa dos vectores de resposta para eliminar as síndromas existentes. Na literatura existente sobre o estado da arte em matéria de algoritmos para geração de vectores de teste não são apresentados procedimentos automáticos que realizem tal análise.

Neste capítulo apresenta-se um conjunto de rotinas, descritas utilizando uma pseudo-linguagem de programação, que correspondem à implementação dos algoritmos adaptativos. Faz-se na secção 5.2 a apresentação das várias fases que compõem uma operação de diagnóstico de faltas, e nas secções seguintes, para cada uma dessas fases, são descritas as rotinas de geração e análise de vectores. O algoritmo dos C vectores óptimos, pela sua completa capacidade de diagnóstico recorrendo a um reduzido número de vectores, ocupa um lugar de destaque entre os restantes algoritmos adaptativos. Neste capítulo é exposta uma proposta de alteração a este algoritmo que tem como fim diminuir a complexidade de análise de síndromas e o conseqüente esforço computacional.

5.1 Introdução

Uma operação de diagnóstico requer a utilização de um equipamento de teste que possibilite a troca de informação entre o local onde são gerados os vectores de teste, que tanto podem ter origem externa como serem gerados no próprio testador, e a CCI a que se destinam.

Na descrição formal que se vai efectuar utilizam-se estruturas de dados do tipo vector para armazenar a informação mais importante do ciclo de teste: VTSs, os vectores de teste a aplicar a cada ligação através da(s) cadeia(s) *boundary scan*, e os VRSs devolvidos pela carta. Recorde-se que os vectores de teste paralelos são aplicados à CCI em formato série, sendo da responsabilidade do testador usado a serialização dos dados [FAP+91]. Por motivos de simplicidade usar-se-ão vectores sequenciais e não vectores paralelos, por se gerarem todos os *bits* de uma só vez para cada ligação da CCI. No entanto, sendo necessário, é fácil passar de um formato para o outro.

As operações sobre VTSs e VRSs exigem a manipulação frequente dos *bits* que os constituem. Para esse efeito supõem-se existentes as subrotinas *bit(num, pos)*, *set(num, pos)* e *reset(num, pos)*. A primeira é usada para testar o valor do *bit* na posição *pos* do número decimal *num* correspondente a um vector de teste/resposta. As subrotinas *set(num, pos)* e *reset(num, pos)* permitem fixar o valor do *bit* na posição *pos* a 1 e a 0, respectivamente.

5.2 Procedimento de teste e diagnóstico

Uma operação de diagnóstico de faltas existentes numa CCI, começa pela geração dos vectores de teste a aplicar através de um testador, ao que se segue a leitura das respostas fornecidas pela CCI. Da análise destas respostas, segundo um algoritmo específico, extrai-se a informação possível acerca das faltas existentes. A utilização de algoritmos adaptativos conduz a melhores resultados que os algoritmos de um só passo quando é exigida uma completa capacidade de diagnóstico, conforme foi concluído no capítulo anterior. A geração de vectores de teste em dois passos, típica dos algoritmos adaptativos, permite uma repartição do algoritmo em fases, conforme se ilustra no fluxograma da figura 5.1.

O primeiro passo decompõe-se em três fases:

1. Geração do primeiro conjunto de VTSs, úteis para efeitos de detecção de faltas.
2. Aplicação deste conjunto de vectores através da(s) cadeia(s) BST existente(s) na CCI e recepção das respostas com a informação necessária à fase seguinte.
3. Detecção de faltas através da identificação das síndromas eventualmente existentes.

No final do primeiro passo é possível concluir se é necessário ou não o segundo passo do algoritmo, pois caso não tenham sido detectadas síndromas o diagnóstico fica completo. Caso o segundo passo seja então necessário, e será esta a situação mais comum quando existirem curto-circuitos, este pode decompor-se em três fases idênticas às já estabelecidas para o primeiro passo.

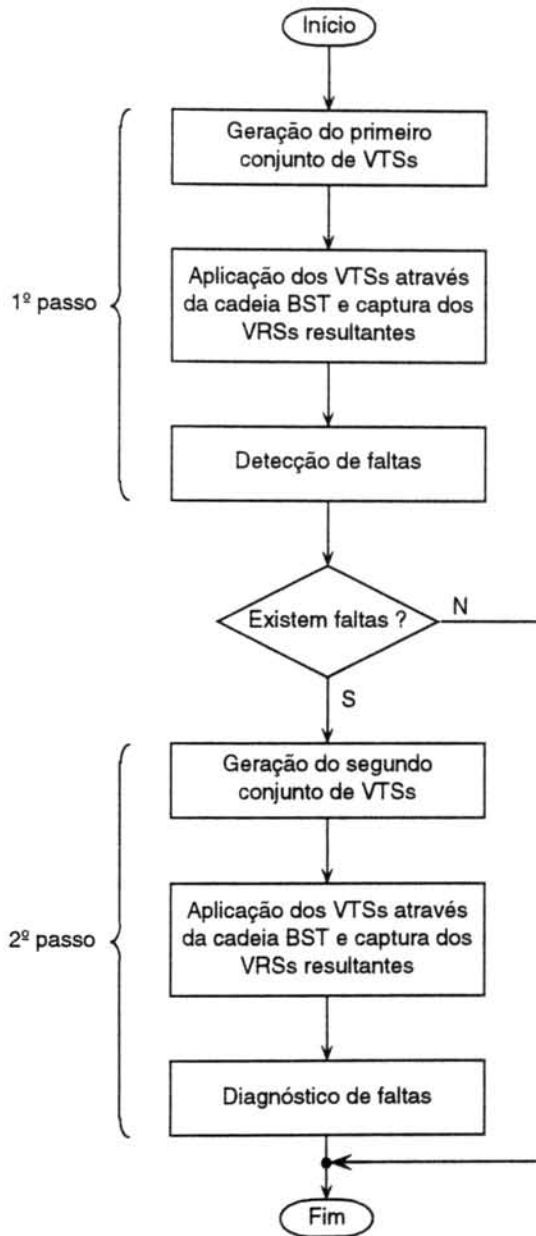


Figura 5.1: Metodologia de diagnóstico usando um algoritmo adaptativo.

Nas secções seguintes incluem-se as fases de geração do primeiro conjunto de VTPs, detecção de faltas, geração do segundo conjunto de VTPs e diagnóstico de faltas. As outras duas fases incluem operações de interacção com a CCI a testar, específicas do testador usado para tal, pelo que se excluem desta descrição. Para descrever as principais rotinas utiliza-se uma pseudo-linguagem de programação, que pela sua simplicidade não exige nenhuma referência prévia.

5.2.1 Geração do primeiro conjunto de vectores de teste

A geração do primeiro conjunto de VTSs caracteriza-se pela sua simplicidade para os quatro algoritmos que se indicam:

- algoritmo dos W vectores
- algoritmo do vector único
- algoritmo dos C-1 vectores
- algoritmo do diagnóstico em dois passos

Para estes algoritmos o VTS a gerar para uma dada ligação tem o valor decimal que identifica a própria ligação. Note-se apenas que há uma ligeira diferença para o último algoritmo mencionado, pois este inclui o VTS nulo. A rotina de geração deste primeiro conjunto de vectores resulta extremamente simples como se indica a seguir.

```
/*
  Geração do primeiro conjunto de VTSs.
*/
para lig ← 1 até N
  VTS[lig] ← lig;
fim.
```

Os algoritmos A1 e A2 geram VTSs bastante mais difíceis de obter, correspondendo aos gerados pelo algoritmo do auto-diagnóstico. A primeira tarefa consiste em determinar o menor número de VTPs necessários ao teste. De seguida obtém-se o primeiro VTS atendendo a que apresenta $\lfloor p/2 \rfloor$ bits 0 e os restantes iguais a 1. Os restantes $N - 1$ VTSs geram-se a partir do primeiro por manipulação dos bits constituintes conforme foi já apresentado na secção 3.3.1.7.

```

/*
  Geração do primeiro conjunto de VTSS.
*/
p ← 1;
enquanto  $\binom{p}{\lfloor p/2 \rfloor} < N$ 
  p ← p + 1;
num_zeros ←  $\lfloor p/2 \rfloor$ ;
/* Geração do primeiro VTS. */
VTS[1] ← 0;
para i ← num_zeros até p - 1
  set(VTS[1], i);
/* Geração dos restantes N - 1 VTSS. */
para lig ← 2 até N
{
  VTS[lig] ← VTS[lig - 1];
  se bit(VTS[lig], 0) = 1 então
  {
    reset(VTS[lig], 0);
    i ← 1;
    cont ← 1;
    enquanto bit(VTS[lig], i) = 1
    {
      reset(VTS[lig], i);
      cont ← cont + 1;
      i ← i + 1;
    }
    i ← i + 1;
    enquanto (bit(VTS[lig], i) = 0 ∧ i < p)
      i ← i + 1;
    se i < p então
    {
      reset(VTS[lig], i);
      j ← i - 1;
      enquanto j > i - cont - 2
      {

```

```

        set(VTS[lig], j);
        j ← j - 1;
    }
}
senão
{
    i ← 1;
    enquanto (bit(VTS[lig], i) ≠ 1 ∧ i < p)
        i ← i + 1;
    reset(VTS[lig], i);
    set(VTS[lig], i - 1);
}
}
fim.

```

5.2.2 Detecção de faltas

A rotina seguinte permite identificar as síndromas do tipo *aliasing* para o algoritmo dos W vectores. Percorrendo as N ligações assinalam-se numa lista aquelas cujo VRS difere do VTS aplicado.

```

/*
    Detecção de faltas através da identificação de síndromas.
*/
W ← 0;
para lig ← 1 até N
    se VRS[lig] ≠ VTS[lig] então
    {
        W ← W + 1;
        lista[W] ← lig; /* Lista de ligações faltosas. */
        falta[lig] ← 1;
    }
senão
    falta[lig] ← 0;
fim.

```

Para o algoritmo do vector único a identificação das síndromas do tipo *aliasing*, é ligeiramente mais elaborada, pois às ligações não defeituosas cujo VRS é igual ao VRS de ligações com faltas terá de ser atribuído um 0, enquanto às restantes é aplicado um 1, no segundo passo.

```

/*
  Detecção de síndromas do tipo aliasing.
*/
alias ← 0; /* Contador de aliasing. */
para lig ← 1 até N
  aliasing[lig] ← 0
para lig1 ← 1 até N
{
  falta[lig1] ← 0
  se VRS[lig1] = VTS[lig1] então
    para lig2 ← 1 até N
      se (lig2 ≠ lig1 ∧ VRS[lig2] ≠ VTS[lig2] ∧ VRS[lig2] = VRS[lig1]) então
        {
          se alias = 0 então
            lista[alias] ← lig1
          senão
            se lista[alias] ≠ lig1 então
              {
                alias ← alias + 1
                lista[alias] ← lig1
              }
            aliasing[lig2] ← lig1
            falta[lig1] ← 1
        }
}
}
fim.

```

A rotina que se segue tem aplicação no algoritmo do diagnóstico em dois passos. Este algoritmo apresenta a particularidade de não fazer depender o segundo conjunto de vectores das respostas ao primeiro conjunto. Contudo, o segundo passo só é aplicado se necessário, isto é, se for detectada alguma falta. Para isso, basta percorrer as N ligações e fazer tal verificação.

```

/*
  Percorre as N ligações até encontrar uma falta. Se não
  encontrar nenhuma então o segundo conjunto de vectores de teste
  não é necessário.
*/
lig ← 1;
enquanto (lig ≤ N ∧ VRS[lig] = VTS[lig])
  lig ← lig + 1;
se lig ≤ N então /* Foi detectada uma falta. */
  falta ← 1;
senão
  falta ← 0;
fim.

```

5.2.3 Geração do segundo conjunto de vectores de teste

O segundo passo de alguns algoritmos utiliza uma estratégia idêntica à do algoritmo da sequência caminhante, aplicando vectores com o elemento caminhante 0 às ligações faltosas detectadas na fase anterior. A rotina que se segue aplica-se ao algoritmo dos W vectores e ao algoritmo A1 e pode também ser adoptada por A2.

```

/*
  Geração do segundo conjunto de VTSs.
*/
zero_caminhante ←  $2^{W-1} - 1$ ;
uns ←  $2^W - 1$ ; /* Número que em binário é constituído por W bits 1. */
deslocamento ← 0;
para lig ← 1 até N
  se falta[lig] = 1 então
  {
    VTS[lig] ← zero_caminhante;
    deslocamento ← deslocamento + 1;
    aux ← W - deslocamento;
    zero_caminhante ← zero_caminhante +  $2^{aux} - 2^{aux-1}$ ;
  }
senão

```

```

VTS[lig] ← uns;
fim.

```

A seguinte rotina gera o único VTP previsto no algoritmo do vector único, considerando a informação disponibilizada pela fase anterior.

```

/*
  Geração do único VTP do segundo conjunto.
*/
para lig ← 1 até N
  se falta[lig] = 1 então
    VTS[lig] ← 0
  senão
    VTS[lig] ← 1
fim.

```

Recorde-se que os VTSs gerados no segundo passo pelo algoritmo do diagnóstico em dois passos traduzem o número de *bits* 0 contidos nos VTPs correspondentes do primeiro passo. A rotina seguinte implementa precisamente esta codificação.

```

/*
  Geração do segundo conjunto de VTSs.
*/
p ← ⌈log2 N⌉;
para lig ← 1 até N
{
  num_zeros ← 0;
  para pos ← 0 até p - 1
  {
    aux ← bit(lig - 1, pos); /* VTS[lig - 1] ≡ (lig - 1) */
    se aux = 0 então
      num_zeros ← num_zeros + 1;
  }
  VTS[lig] ← num_zeros;
}
fim.

```

5.2.4 Diagnóstico de faltas

As duas últimas rotinas que se incluem, referem-se ao diagnóstico de faltas do tipo curto-circuito efectuado pelos algoritmos dos W vectores e do vector único, respectivamente. Consistem essencialmente na eliminação de síndromas dos tipos *aliasing* e *confounding* para apurar quais os curto-circuitos que de facto ocorrem. Para esse fim pesquisam-se as ligações usando a informação recolhida durante a detecção.

```

/*
    Diagnóstico de faltas.
*/
para  $i \leftarrow 1$  até  $W$ 
    para  $j = i + 1$  até  $W$ 
        se  $VRS[lista[i]] = VRS[lista[j]]$  então
            escreve("Existe um curto-circuito entre as
                ligações ",  $i$ , " e ",  $j$ );
fim.

/*
    Diagnóstico de faltas para as situações de aliasing
    previamente detectadas.
*/
para  $lig \leftarrow 1$  até  $N$ 
    se  $VRS[lig] \neq VTS[lig]$  então
        {
             $aux \leftarrow aliasing[lig]$ ;
            se  $VRS[lig] = VRS[aux]$  então
                escreve("Existe um curto-circuito entre as
                    ligações ",  $i$ , " e ",  $j$ );
        }
fim.

```

Foi reservado para discussão na secção seguinte o algoritmo dos C vectores óptimos e a respectiva descrição formal.

5.3 Alteração ao algoritmo dos C vectores óptimos

Nesta secção revisita-se o algoritmo dos C vectores óptimos com os objectivos de analisar com detalhe a complexidade da geração do segundo conjunto de vectores de teste, e de propor uma solução para minimizar essa complexidade, alterando o modo de gerar os referidos VTPs.

5.3.1 Análise do segundo passo do algoritmo

Relembrando o que foi exposto na secção 3.3.2.3, $C - 1$ VTPs são suficientes para resolver as situações de *confounding*. Se também existirem síndromas do tipo *aliasing*, estas poderão ser ultrapassadas recorrendo a um vector adicional, conforme exposto para o algoritmo do vector único (secção 3.3.2.2). Descreve-se de seguida a metodologia de geração dos $C - 1$ vectores.

O primeiro VTP é gerado atribuindo um 0 a uma das ligações envolvidas na primeira falta do número máximo de faltas consideradas independentes, sendo isto feito para todas as síndromas do tipo *confounding* existentes, e às restantes ligações é atribuído um 1. O segundo VTP é gerado atribuindo um 0 a uma das ligações envolvidas na segunda falta do número máximo de faltas consideradas independentes, e mais uma vez, isto é feito para todas as faltas do tipo *confounding*, e às restantes ligações é atribuído um 1. Este processo é repetido até o $(C - 1)$ -ésimo, e último vector, ser gerado. Como C é o maior grau de *confounding* presente, este processo de solucionar as síndromas do tipo *confounding* termina primeiro para as síndromas de menor grau. Quando isto acontece, a todas as ligações envolvidas nestas síndromas é atribuído um 1 nos VTPs seguintes. Quando o $(C - 1)$ -ésimo VTP é capturado, o conjunto de VTPs obtido deste modo é suficiente para concluir se a última falta considerada independente é realmente independente ou não, sendo deste modo necessários $C - 1$ VTPs apenas.

Repare-se que para cada síndrome do tipo *confounding*, de grau c , terão que ser considerados todos os subconjuntos de ligações que produzam o mesmo VRS, a partir dos quais se determinarão os que correspondem aos curto-circuitos independentes que verifiquem a condição de manter o próprio VRS da síndrome [JY89]. Numa situação genérica, poderá existir mais do que uma combinação de c subconjuntos de ligações nestas circunstâncias. Sendo assim, todas estas possibilidades terão que ser analisadas, para permitir a formação de um subconjunto com c ligações, que terá a propriedade de conter pelo menos uma das ligações envolvidas em cada um dos subconjuntos considerados. É precisamente nesta tarefa que reside a principal dificuldade deste algoritmo, pois determinar este conjunto de

c ligações para cada síndrome do tipo *confounding* presente, constitui um problema de cobertura de conjuntos, NP-completo [Eve79], [GJ79], [WMR90].

A tabela 5.1 ilustra um exemplo que esclarece a utilização deste algoritmo no que respeita à eliminação de situações de *confounding*. Para isso, foi criada propositadamente uma situação em que são originadas duas síndromas diferentes do tipo *confounding*, por curto-circuitos existentes entre as ligações $\langle 4, 12 \rangle$, $\langle 5, 14 \rangle$, $\langle 6, 21 \rangle$, $\langle 17, 18 \rangle$ e $\langle 19, 20 \rangle$.

Ligação	Primeiro passo			Segundo passo				
	VTS	VRS	<i>confounding</i>	VTP1	VRP1	VTP2	VRP2	
1	00001	00001	-					
2	00010	00010	-					
3	00011	00011	-					
*	4	00100	00100	c_1	0	0	1	1
★	5	00101	00100	c_1	1	1	0	0
◇	6	00110	00100	c_1	1	1	1	1
7	00111	00111	-					
8	01000	01000	-					
9	01001	01001	-					
10	01010	01010	-					
11	01011	01011	-					
*	12	01100	00100	c_1	1	0	1	1
13	01101	01101	-					
★	14	01110	00100	c_1	1	1	1	0
15	01111	01111	-					
16	10000	10000	-					
●	17	10001	10000	c_2	0	0	1	1
●	18	10010	10000	c_2	1	0	1	1
○	19	10011	10000	c_2	1	1	1	1
○	20	10100	10000	c_2	1	1	1	1
◇	21	10101	00100	c_1	1	1	1	1

Tabela 5.1: Eliminação de síndromas do tipo *confounding* pelo algoritmo dos C vectores óptimos.

A síndrome do tipo *confounding* c_1 envolve as ligações 4, 5, 6, 12, 14 e 21, que dão origem ao VRS 00100, e tem grau 3. A síndrome c_2 envolve as ligações 17, 18, 19 e 20, que dão origem ao VRS 10000, e tem grau 2. Como $C = c_1$, o segundo conjunto de VTPs é constituído por 2 vectores que eliminam completamente as duas síndromas do tipo *confounding*. Por motivos de simplicidade, na tabela 5.1 apenas foram incluídos os VTSs e VRSs referentes às ligações com *confounding*.

O algoritmo dos C vectores óptimos, conforme formulado em [JY89], permite gerar o primeiro VTP sem qualquer dificuldade, conforme o exemplo apresentado mostra: foi atribuído um 0 à primeira ligação de cada subconjunto de ligações que respondem com o mesmo VRS (ligações 4 e 17), e um 1 a todas as restantes. Contudo, a geração do segundo VTP já é problemática: a que ligação se deve aplicar o 0 com o objectivo de acabar de eliminar a situação de *confounding* c_1 ? Será indiferente, desde que não coincida com a ligação considerada no primeiro VTP? A resposta a esta questão é negativa. De facto, se o 0 for aplicado à ligação 12, a síndrome c_1 fica por resolver, e note-se que ao gerar o segundo VTP, o algoritmo não tem conhecimento da resposta ao primeiro VTP, pelo que não se sabe se o curto-circuito independente que envolve a ligação 12 já foi identificado. Se em vez de $C = 3$, o grau de *confounding* for superior, este problema irá colocar-se em todos os $C - 2$ VTPs que sucedem o primeiro.

5.3.2 Proposta de uma solução

Uma solução para ultrapassar esta importante dificuldade consiste em tornar a geração do segundo conjunto de vectores interactiva, no sentido de fazer depender cada um dos $C - 2$ últimos vectores, da identificação de curto-circuitos independentes permitida pelo VTP anterior. Isto exige a aplicação de cada VTP gerado no segundo passo e a recolha do respectivo VRP, o qual permitirá eliminar mais uma ligação em cada subconjunto de ligações caracterizado pela mesma síndrome, à qual não deve ser aplicado um 0. Repare-se que no exemplo apresentado na tabela 5.1 o único 0 foi atribuído à ligação 5, pois já era conhecido do primeiro VRP que não se devia aplicar à ligação 12. Esta solução considera pois a divisão do segundo passo em $C - 1$ passos, cada um deles constituído pela geração dum vector, destinados à eliminação de *confoundings*.

Se por um lado esta abordagem simplifica a complexidade inerente ao segundo passo deste algoritmo adaptativo, por outro lado a aplicação dos VTPs através da infra-estrutura BST será mais demorada, pois cada um dos VTPs gerados deve reflectir o conhecimento proporcionado pelo VTP antecedente. Contudo, esta desvantagem pode perder importância comparada com a de requisitos de diagnóstico exigentes, em situações onde o critério predominante seja a resolução do diagnóstico.

5.3.3 Descrição algorítmica

Apresenta-se de seguida o algoritmo completo, com as quatro principais etapas descritas na secção 5.2, considerando a proposta de alteração que acabou de se descrever. A identificação de situações de *confounding* revela-se neste algoritmo como crucial, pois

depende dela a correcta geração de vectores para efeitos de diagnóstico. Os nomes das variáveis auxiliares utilizadas no programa são auto-explicativos, e as várias tarefas do algoritmo completo encontram-se identificadas com comentários.

```

/*
  Geração do primeiro conjunto de vectores de teste.
*/
para lig ← 1 até N
  VTS[lig] ← lig;

/*
  Identificação das síndromas do tipo confounding.
*/
num_conf ← 0;
num_lig ← 1;
para lig1 ← 1 até N
  aux[lig1] ← 0; /* Confounding ainda não detectado. */
para lig1 ← 1 até N - 1
  se aux[lig1] = 0 então
  {
    cont ← 1;
    i ← lig1 + 1;
    para lig2 ← i até N
      se VRS[lig2] = VRS[lig1] então
        cont ← cont + 1;
    se cont ≥ 4 então
    {
      num_conf ← num_conf + 1;
      grau[num_conf] ← [cont/2]; /* Grau do confounding encontrado. */
      conf[num_lig] ← lig1; /* Ligações com confounding. */
      para lig2 ← i até N
        se VRS[lig2] = VRS[lig1] então
        {
          num_lig ← num_lig + 1;
          conf[num_lig] ← lig2;
          aux[lig2] ← 1;

```

```

    }
  }
}

/*
  Cálculo do maior grau de confounding.
*/
C ← 0;
para i ← 1 até num_conf
  se grau[i] > C então
    C ← grau[i];

/*
  Identificação das síndromas do tipo aliasing.
*/
alias ← 0; /* Contador de aliasing. */
para lig ← 1 até N
  aliasing[lig] ← 0
para lig1 ← 1 até N
{
  falta[lig1] ← 0
  se VRS[lig1] = VTS[lig1] então
    para lig2 ← 1 até N
      se (lig2 ≠ lig1 ∧ VRS[lig2] ≠ VTS[lig2] ∧ VRS[lig2] = VRS[lig1]) então
        {
          se alias = 0 então
            lista[alias] ← lig1
          senão
            se lista[alias] ≠ lig1 então
              {
                alias ← alias + 1
                lista[alias] ← lig1
              }
            aliasing[lig2] ← lig1
            falta[lig1] ← 1
        }
}
}

```

```

}

/*
  Apuramento de faltas nas situações de aliasing
  anteriormente encontradas.
*/
para lig ← 1 até N
  se VRS[lig] ≠ VTS[lig] então
  {
    aux ← aliasing[lig];
    se VRS[lig] = VRS[aux] então
      escreve("Existe um curto-circuito entre as
              ligações ", lig, " e ", aux);
  }

/*
  Segundo passo do algoritmo: geração interactiva de VTPs e análise
  de VRPs para eliminação das situações de confounding.
*/
para lig ← 1 até N
  id[lig] ← 0; /* Assinalam-se com 0 as ligações ainda não testadas. */
para k ← 1 até C - 1
  {
    para lig ← 1 até N
      VTS[lig] ← 1;
    l ← 1;
    para i ← 1 até num_conf
      {
        lig ← conf[l];
        enquanto id[lig] = 0
          lig ← lig + 1;
        n ← 2 * grau[i]; /* Número de ligações que originam o confounding i. */
        l ← l + n; /* Posição ocupada em conf pela primeira ligação. */
        VTS[lig] ← 0;
      }
  }

```

```

/*
  Aplica o VTP gerado através da infraestrutura BST.
*/
aplicaVTP;
/*
  Recolhe o VRP proveniente da infraestrutura BST.
*/
recolheVRP;

l ← 1;
para i ← 1 até num_conf
{
  n ← 2 * grau[i];
  para lig1 ← l até l + n
    se id[lig1] ≠ 0 então
      {
        lig2 ← conf[l];
        se id[lig2] ≠ 0 então
          se VRS[lig2] = VRS[lig1] então
            {
              escreve("Existe um curto-circuito
                entre as ligações ", lig1, " e ", lig2);
              id[lig1] ← 0;
              id[lig2] ← 0;
            }
          }
        }
    l ← l + n;
  }
}

/*
  Diagnóstico do último curto-circuito na(s) síndrome(s)
  do tipo confounding de maior grau.
*/
l ← 1;
para i ← 1 até num_conf

```

```

{
  n ← 2 * grau[i];
  para lig1 ← l até l + n - 1
    para lig2 ← l + 1 até l + n
      se (id[lig2] ≠ 0 ∧ id[lig1] ≠ 0) então
        escreve("Existe um curto-circuito entre as
                ligações ", lig2, " e ", lig1);
}
fim.

```

A etapa final deste programa é necessária para identificar o par de ligações que corresponde ao último curto-circuito existente e que pertence ao grupo de ligações com o maior grau de *confounding*, pois o algoritmo só gera $C - 1$ vectores. O processo utilizado pesquisa na mesma todos os *confoundings* diferentes porque na realidade pode haver mais do que um subconjunto de ligações com o mesmo grau de *confounding* máximo.

5.4 Conclusão

A análise detalhada e exaustiva contida nos capítulos precedentes permitiu formalizar a codificação dos procedimentos de detecção e diagnóstico incluídos neste capítulo, além de possibilitar também que tenha sido abordada uma solução para baixar a complexidade de análise de síndromas no algoritmo dos C vectores óptimos.

Capítulo 6

Conclusão

Para ser possível detectar e localizar faltas em cartas de circuito impresso é necessário em primeiro lugar tornar possível o acesso às entradas a testar. No capítulo 2 fez-se uma descrição das duas técnicas de teste mais implantadas na indústria de componentes e sistemas electrónicos, o teste *in-circuit* e o teste funcional. Estas técnicas de teste, com as suas vantagens e desvantagens relativas, têm satisfeito as necessidades dos fabricantes. Porém, o rápido desenvolvimento que se tem verificado nas tecnologias de fabricação dos componentes tem levantado crescentes problemas aos tradicionais equipamentos de teste, quer devido à grande complexidade atingida pelos componentes quer devido às novas tecnologias de montagem superficial e de encapsulamento. O aparecimento do teste baseado no *boundary scan* veio responder aos principais problemas apresentando uma solução electrónica normalizada, com níveis de aceitação elevados por um grande número de fabricantes de componentes.

Outra consequência que o aparecimento do BST teve foi tornar possível o desenvolvimento de equipamentos de teste de baixo custo, sem perda de desempenho relativamente a equipamentos tradicionais de teste com custos mais elevados. O capítulo 3 descreveu numa forma sistemática e exaustiva as várias formas algorítmicas de abordar o problema de geração de testes para CCIs. Foi dada grande importância ao compromisso entre a dimensão dos testes gerados por cada estratégia e a capacidade de detecção e diagnóstico atingíveis. Para um cenário de teste exigente do ponto de vista de diagnóstico concluiu-se que a classe dos algoritmos adaptativos era a que melhor satisfazia as expectativas e requisitos colocados. Além desse aspecto, mostrou-se também que cumpriam com bons resultados a detecção de faltas utilizando somente a geração de testes proporcionada pelo primeiro passo, caracterizada por um reduzido número de vectores de teste.

O desenvolvimento de um equipamento de teste automático com capacidades de diagnóstico

implica a necessidade de desenvolvimento de *software* para a concretização desta tarefa. A primeira etapa desse desenvolvimento passa necessariamente por um estudo prévio das alternativas funcionais de algoritmos de teste e diagnóstico expostas no capítulo 3. A fase seguinte exige a elaboração de uma especificação formal antes do desenvolvimento da aplicação de *software*. O capítulo 5 descreu um procedimento de diagnóstico, decomposto na geração de vectores de teste, detecção de faltas, geração de vectores adicionais e diagnóstico de faltas. A especificação formal destas principais etapas do procedimento foi feita para os algoritmos adaptativos, utilizando para tal efeito uma pseudo-linguagem de programação simples.

Nesse mesmo capítulo foi apresentada e discutida ainda uma alteração à forma de implementação do algoritmo adaptativo dos C vectores óptimos, que é porventura aquele que melhores condições reúne para enfrentar cenários de diagnóstico exigentes. A alteração introduzida teve como objectivo simplificar a eliminação de ambiguidades no diagnóstico de faltas do tipo curto-circuito.

Uma evolução natural das especificações descritas aponta no sentido de uma implementação concreta, com base num testador, que possa comprovar a eficácia realçada para os algoritmos e que ao mesmo tempo sirva de meio experimental para alcançar novos resultados. Para este fim poderá vir a ser utilizado um testador desenvolvido no INESC-Porto pelo grupo Projecto Testável de Sistemas Electrónicos.

Apêndice A

Apêndice

A.1 Independência diagonal dum conjunto de vectores

Seja $B_{n \times m}$ a matriz que se obtém dispendo, segundo as n linhas os N VTSs pertencentes a um conjunto de vectores de teste, resultando m colunas que se identificam com os p VTPs. Considere-se $m \geq n$ e b_{ij} , onde $0 \leq i \leq n - 1$ e $0 \leq j \leq m - 1$, um elemento de B . Se B , ou a matriz obtida desta por trocas sucessivas de linhas e/ou colunas, é tal que

$$b_{ij} = \begin{cases} 0 & \Leftarrow i = j \\ 1 & \Leftarrow i > j \\ x & \Leftarrow i < j \end{cases} \quad (\text{A.1})$$

onde $x \in \{0, 1\}$ (valor *don't care*), então B diz-se *diagonalmente independente*. A troca de linhas ou colunas de B altera os VTSs aplicados às ligações. A troca de linhas apenas afecta os VTSs aplicados às correspondentes ligações, enquanto a troca de colunas pode afectar os VTSs de todas as ligações. Contudo, a troca de colunas corresponde a alterar a ordem pela qual os VTPs são aplicados. Assim sendo, os fundamentos do procedimento de geração dos vectores de teste não são afectados, pelo que as trocas de linhas e/ou colunas de B são permitidas.

Mostra-se de seguida que para uma matriz de vectores de teste diagonalmente independente, não ocorrem síndromas do tipo *aliasing* ou *confounding*. Uma síndrome do tipo *aliasing* ocorre se os VTSs correspondentes às ligações em curto-circuito geram *bits* 0 nas mesmas posições, em que existam também *bits* 0, do VTS correspondente à ligação que não possua faltas. Esta situação nunca acontece quando a matriz dos vectores de teste goza da propriedade da independência diagonal, pois para isso acontecer teriam de existir pelo menos dois VTSs com o primeiro 0 na mesma posição. Similarmente, não

existem síndromas do tipo *confounding*, pois isso só poderia acontecer se dois conjuntos independentes de ligações curto-circuitadas produzissem o mesmo VRS. Para uma matriz de vectores de teste que verifique a propriedade da independência diagonal, não é possível existirem dois conjuntos de vectores nos quais exista pelo menos um vector em cada conjunto que possua o primeiro 0 na mesma posição.

Fica assim provada a completa capacidade de diagnóstico dum algoritmo que dê origem a uma matriz de vectores de teste, que verifica a propriedade da independência diagonal.

Como exemplos de matrizes possuindo a propriedade da independência diagonal, vejamos as que se seguem, em que M_C corresponde à matriz obtida pelo algoritmo da sequência caminhante para $N = 4$.

$$M_A = \begin{bmatrix} 0 & x & x & x & x & x \\ 1 & 0 & 1 & 0 & 0 & x \\ 1 & 1 & 0 & x & x & 1 \\ 1 & 1 & 1 & 0 & x & x \end{bmatrix}$$

$$M_B = \begin{bmatrix} 0 & x & x & 1 & 1 & 1 \\ 1 & 0 & x & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & x & 0 & 1 \\ 1 & 1 & 1 & x & x & 0 \end{bmatrix}$$

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

Note-se que M_B verifica de facto a independência diagonal, pois a expressão A.1 é verificada trocando as linhas 4 e 6, trocando depois as colunas 4 e 6, se $x = 0$.

A.2 VTPs gerados pelo algoritmo da independência máxima

Deduz-se nesta secção a expressão 3.8 que permite obter o número de VTPs necessários para que a capacidade de diagnóstico do algoritmo da independência máxima seja completa. Recorde-se que p depende da extensão E do curto-circuito que envolve o maior número de ligações, de entre todos os possíveis curto-circuitos, e do número de ligações N da CCI, supondo que estas se encontram ordenadas por adjacência. Os conceitos de ordenação por adjacência de um conjunto de ligações, extensão dum curto-circuito e peso potencial dum VTS, foram introduzidos na secção 3.3.1.6.

A tabela A.1 mostra como a extensão e dum curto-circuito (coluna 2) depende do peso potencial \bar{w} e do número de VTPs p . Na coluna 3 da mesma tabela indica-se o número de VTSs $N_{\bar{w}}$ (relembre-se aqui a expressão 3.6) que se podem obter com p bits e com um peso potencial \bar{w} , excluindo os VTSs $00 \dots 0$ e $11 \dots 1$ para que as faltas do tipo fixo-a sejam detectadas. As características indicadas na tabela A.1 supõem a garantia do diagnóstico total de faltas por parte do algoritmo da independência máxima.

\bar{w}_i	e_i	$N_{\bar{w}}$
1	p	p
2	p	$(p - 1)2^0$
3	$p - 1$	$(p - 2)2^1$
4	$p - 2$	$(p - 3)2^2$
\vdots	\vdots	\vdots
$k - 1$	$p - (k - 3)$	$[p - (k - 2)]2^{k-3}$
k	$p - (k - 2)$	$[p - (k - 1)]2^{k-2}$

Tabela A.1: Características dos VTSs gerados pelo algoritmo da independência máxima.

À medida que o peso potencial \bar{w}_i dum subconjunto de VTSs aumenta, diminui a extensão máxima e_i que um curto-circuito pode ter, mantendo a total capacidade de diagnóstico do algoritmo. Assim, para o valor máximo $\max\{\bar{w}_i\} = k$, a extensão máxima de curto-circuitos E que o conjunto de vectores de teste garante diagnosticar é $\min\{e_i\} = p - (k - 2)$.

O número total de VTSs gerados pelo algoritmo da independência máxima deve ser igual ao número de ligações N da CCI, isto é, deve ser igual à soma dos termos presentes na coluna 3 da tabela A.1. Assim,

$$\begin{aligned}
 N &= p + p \sum_{i=0}^{k-2} 2^i - \sum_{i=0}^{k-2} (i+1)2^i \\
 &= p + p \sum_{i=0}^{k-2} 2^i - \sum_{i=0}^{k-2} i2^i - \sum_{i=0}^{k-2} 2^i \\
 &= p + (p-1) \sum_{i=0}^{k-2} 2^i - \sum_{i=1}^{k-2} i2^i
 \end{aligned} \tag{A.2}$$

Das progressões geométricas é conhecida a expressão A.3.

$$\sum_{i=1}^n r^i = u_1 \frac{1-r^n}{1-r} \tag{A.3}$$

Aplicando este resultado ao cálculo da segunda parcela da expressão A.2, com $r = 2$, vem

$$\begin{aligned}
 N &= p + (p-1) \frac{1-2^{k-1}}{1-2} - \underbrace{\sum_{i=1}^{k-2} i2^i}_{\mathcal{S}} \\
 &= (p-1)2^{k-1} + 1 - \underbrace{\sum_{i=1}^{k-2} i2^i}_{\mathcal{S}}
 \end{aligned} \tag{A.4}$$

Veja-se agora como se poderá calcular a parcela \mathcal{S} da expressão A.4. Para tal, considere-se a seguinte soma de potências:

$$\begin{array}{ccccccc}
 a^1 & + & & & & & \\
 a^2 & + & a^2 & + & & & \\
 a^3 & + & a^3 & + & a^3 & + & \\
 \vdots & & \vdots & & \vdots & & \ddots \\
 \underbrace{a^n}_{\mathcal{S}_1} & + & \underbrace{a^n}_{\mathcal{S}_2} & + & \underbrace{a^n}_{\mathcal{S}_3} & + & \cdots + \underbrace{a^n}_{\mathcal{S}_n}
 \end{array}$$

Como $\mathcal{S}_1 = \sum_{i=1}^n a^i$, $\mathcal{S}_2 = \sum_{i=2}^n a^i$, $\mathcal{S}_3 = \sum_{i=3}^n a^i$ e $\mathcal{S}_n = \sum_{i=n}^n a^i$, então

$$\sum_{i=1}^n ia^i = \sum_{i=1}^n \sum_{j=i}^n a^j \tag{A.5}$$

No caso presente, temos $a = 2$, pelo que utilizando a expressão A.5, a parcela \mathcal{S} da expressão A.4 toma o seguinte aspecto:

$$\begin{aligned}
 S &= \sum_{i=1}^{k-2} i2^i \\
 &= \sum_{i=1}^{k-2} \sum_{j=i}^{k-2} 2^j \\
 &= \sum_{i=1}^{k-2} 2^i \frac{1 - 2^{k-i-1}}{1 - 2} \\
 &= \sum_{i=1}^{k-2} (2^{k-1} - 2^i) \\
 &= (k-2)2^{k-1} - \sum_{i=1}^{k-2} 2^i \\
 &= (k-2)2^{k-1} + 2 - 2^{k-1} \\
 &= (k-3)2^{k-1} + 2
 \end{aligned}$$

Calculado o valor de S , resulta finalmente a seguinte expressão para N :

$$\begin{aligned}
 N &= (p-1)2^{k-1} + 1 - (k-2)2^{k-1} - 2 + 2^{k-1} \\
 &= p2^{k-1} - [(k-2)2^{k-1} + 1] \\
 &= [p - (k-2)]2^{k-1} - 1
 \end{aligned}$$

ou

$$N + 1 = [p - (k - 2)]2^{k-1} \tag{A.6}$$

Conforme já foi referido, o diagnóstico de todos os curto-circuitos sem ambiguidades é garantido se o valor mínimo de e_i , a extensão máxima permitida para os curto-circuitos, associada aos subconjuntos de VTSs (veja-se a tabela A.1), é igual a E . Como $\min\{e_i\} = p - (k - 2)$

$$E = p - (k - 2) \tag{A.7}$$

Utilizando a expressão A.7 e calculando o logaritmo binário da expressão A.6 vem

$$\lceil \log_2(N + 1) \rceil = \lceil \log_2 E \rceil + k - 1$$

$$= \lceil \log_2 E \rceil + p - E + 1$$

donde se conclui que

$$p = \lceil E + \log_2(N + 1) - \log_2 E - 1 \rceil \quad (\text{A.8})$$

O termo $\log_2(N + 1)$ revela a exclusão do VTS $00 \cdots 0$. O algoritmo da independência máxima evita a atribuição do VTS $11 \cdots 1$ à ligação n_N [YJ89], desde que

$$E > 2 \vee \log_2(N + 1) < \lceil \log_2(N + 1) \rceil$$

Portanto, quando a condição fronteira for verdadeira, ou seja, quando

$$E = 2 \wedge \log_2(N + 1) = \lceil \log_2(N + 1) \rceil$$

deve subtrair-se uma unidade ao lado direito da expressão A.2, resultando assim

$$p = \lceil E + \log_2(N + 2) - \log_2 E - 1 \rceil \quad (\text{A.9})$$

O termo $\log_2(N + 2)$, presente na expressão A.9, reflecte agora a omissão dos indesejados VTSs $00 \cdots 0$ e $11 \cdots 1$. Finalmente, se $E = 2$, a expressão A.9 resulta em

$$p = \lceil E + \log_2(N + 2) \rceil$$

ficando assim completa a dedução da expressão 3.8 enunciada na secção 3.3.1.6, que se resume a

$$p = \begin{cases} \lceil E + \log_2(N + 1) - \log_2 E - 1 \rceil & \Leftarrow E > 2 \vee \log_2(N + 1) < \lceil \log_2(N + 1) \rceil \\ \lceil \log_2(N + 2) \rceil & \Leftarrow E = 2 \wedge \log_2(N + 1) = \lceil \log_2(N + 1) \rceil \end{cases}$$

A.3 Geração duma sequência caminhante

Seja

$$n = b_{p-1}b_{p-2} \cdots b_{k+1}b_k b_{k-1} \cdots b_1 b_0$$

um número binário constituído por p bits, em que só um deles é 0, o qual ocupa a posição referente ao bit k . No contexto do algoritmo da sequência caminhante, n representa o VTS atribuído à $(p - k)$ -ésima ligação.

Sendo n_1 e n_2 dois números decimais, representando dois VTSs consecutivos, n_2 pode calcular-se a partir de n_1 , o VTS precedente, segundo a expressão A.10.

$$n_1 = 11 \cdots 1011 \cdots 11$$

$$n_2 = 11 \cdots 1101 \cdots 11$$

$$n_2 = n_1 + 2^k - 2^{k-1} ; 1 \leq k \leq p - 1 \quad (\text{A.10})$$

Esta relação deduz-se atendendo à constituição binária dos dois números consecutivos.



Bibliografia

- [ABF90] Miron Abramovici, Melvin A. Breuer, and Arthur D. Friedman. *Digital Systems Testing and Testable Design*. Computer Science Press, 1990.
- [AHU74] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Publishing Company, 1974.
- [AR83] M. Abadir and H. Reghbaty. Functional testing of semiconductor random access memories. *ACM Computing Surveys*, 15:175–198, September 1983.
- [BA84] Dhananjay Brahme and Jacob Abraham. Functional testing of microprocessors. *IEEE Transactions on Computers*, 33(6):475–485, June 1984.
- [Bat85] John Bateson. *In-Circuit Testing*. Van Nostrand Reinhold Company, 1985.
- [Ben82] R. G. Bennetts. *Introduction to Digital Board Testing*. Crane Russak & Company, 1982.
- [Bla83] R. Blahut. *Theory and Practice of Error Control Codes*. Addison-Wesley Publishing Company, 1983.
- [BO91] R. G. Bennetts and A. Osseyran. IEEE standard 1149.1-1990 on boundary scan: History, literature survey, and current status. *Journal of Electronic Test: Theory and Applications*, 2(1):11–26, March 1991.
- [Car93] David H. Carey. Trends in low-cost, high-performance substrate technology. *IEEE Micro*, 13(2):19–27, April 1993.
- [CLW90] Wu-Tung Cheng, James L. Lewandowski, and Eleanor Wu. Diagnosis for wiring interconnects. In *IEEE International Test Conference*, pages 565–571. IEEE Computer Society Press, 1990.
- [Com90] Hewlett-Packard Company. HP boundary scan tutorial and BSDL reference guide, 1990.

- [Dan93] Wayne T. Daniel. IEEE 1149.1 growing pains. In *IEEE International Test Conference*, page 264. IEEE Computer Society Press, 1993.
- [DDD⁺93] C. Dislis, J. H. Dick, I. D. Dear, I. N. Azu, and A. P. Ambler. Economics modelling for the determination of test strategies for complex VLSI boards. In *IEEE International Test Conference*, pages 210–217. IEEE Computer Society Press, 1993.
- [Dem94] Bernard Demarche. In search of the imperfect PCB. *Electronics Specifier International*, 1(1):42–44, 1994.
- [Des88] LSI/VLSI Testability Design. *Frank F. Tsui*. McGraw-Hill Book Company, 1988.
- [dJvdH91] Frans de Jong and Frank van der Heyden. Testing the integrity of the boundary scan test infrastructure. In *IEEE International Test Conference*, pages 106–112. IEEE Computer Society Press, 1991.
- [Eve79] S. Even. *Graph Algorithms*. Computer Science Press, 1979.
- [FAP⁺91] J. M. Ferreira, A. J. Araújo, F. S. Pinto, J. A. Tavares, G. R. Alves, and J. S. Matos. Sistema de validação e teste de cartas de circuito impresso com BST: Especificação da arquitectura global. Relatório de Projecto 2, INESC-Porto, Novembro 1991. JNICT - Proj. No. PMCT/C/TIT/937/90.
- [Fer92] J. M. Ferreira. *O Teste de Cartas de Circuito Impresso com BST: Arquitectura de um Controlador Residente e Geração Automática do Programa de Teste*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, Abril 1992.
- [FMdJ89] José M. Ferreira, José S. Matos, and Frans de Jong. Test pattern generation methodology for full boundary scan test. Technical report, ESPRIT Deliverable D4.1, December 1989. Project 2478 (Research into Boundary Scan Test Implementation).
- [FMP91] J. M. Ferreira, J. S. Matos, and F. S. Pinto. Sistema de validação e teste de cartas de circuito impresso com BST: Definição de requisitos. Relatório de Projecto 1, INESC-Porto, Julho 1991. JNICT - Proj. No. PMCT/C/TIT/937/90.
- [GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

- [GM82] P. Goel and M. McMahon. Electronic chip-in-place test. In *IEEE Test Conference*, pages 83–90. IEEE Computer Society Press, 1982.
- [Gol79] Lawrence H. Goldstein. Controllability/observability analysis of digital circuits. *IEEE Transactions on Circuits and Systems*, 26(9):685–693, September 1979.
- [Gre93] David A. Greene. Benefits of boundary scan to in-circuit test. In *IEEE International Test Conference*, page 263. IEEE Computer Society Press, 1993.
- [Her93] Dennis Herrell. Addressing the challenges of advanced packaging and interconnection. *IEEE Micro*, 13(2):10–18, April 1993.
- [HRA88] Abu Hassan, Janusz Rajski, and Vinod K. Agarwal. Testing and diagnosis of interconnects using boundary scan architecture. In *IEEE International Test Conference*, pages 126–137. IEEE Computer Society Press, 1988.
- [IEE90] IEEE standard test access port and boundary scan architecture, February 1990. IEEE Std. 1149.1.
- [Joh93] Barry Johnson. Boundary scan eases test of new technologies. *Test & Measurement Europe*, 1(1):25–30, 1993.
- [JY89] Najmi Jarwala and Chi W. Yau. A new framework for analyzing test generation and diagnosis algorithms for wiring interconnects. In *IEEE International Test Conference*, pages 63–70. IEEE Computer Society Press, 1989.
- [Kau74] William Kautz. Testing for faults in wiring interconnects. *IEEE Transactions on Computers*, 23(4):358–363, April 1974.
- [LB91] Jung-Cheun Lien and Melvin Breuer. Maximal diagnosis for wiring networks. In *IEEE International Test Conference*, pages 96–105. IEEE Computer Society Press, 1991.
- [LB93] Pradeep Lall and Shrikar Bhagath. An overview of multichip modules. *Solid State Technology*, 36(9):65–76, September 1993.
- [LH88] Chen-Shang Lin and Hong-Fa Ho. Automatic functional test program generation for microprocessors. In *ACM/IEEE Design Automation Conference*, pages 605–608, 1988.
- [Mau92] Colin Maunder. Boundary scan: An end-of-term report. *IEEE Design & Test of Computers*, 9(2):82–85, June 1992.

- [Mau93] Colin Maunder. Position statement: ITC93 boundary scan panel. In *IEEE International Test Conference*, page 262. IEEE Computer Society Press, 1993.
- [MBD91] J. Miles, R. De Bondt, and L. Daemen. A test economics model & cost-benefit analysis of boundary scan. In *IEEE European Test Conference*, pages 375–384, 1991.
- [McC85] Edward J. McCluskey. Built-in self-test techniques. *IEEE Design & Test of Computers*, pages 21–28, April 1985.
- [Mis93] David Misunas. Advanced packaging and interconnection technology. *IEEE Micro*, 13(2):7–9, April 1993.
- [MM93] D. McBean and W. R. Moore. Testing interconnects: A pin adjacency approach. In *IEEE European Test Conference*, pages 484–490, 1993.
- [MT90] Colin M. Maunder and Rodham E. Tulloss. *The Test Access Port and Boundary Scan Architecture*. IEEE Computer Society Press, 1990.
- [MT91] C. M. Maunder and R. E. Tulloss. An introduction to the boundary scan standard: ANSI/IEEE std. 1149.1. *Journal of Electronic Test: Theory and Applications*, 2(1):27–42, March 1991.
- [MT92] Colin M. Maunder and Rodham E. Tulloss. Testability on TAP. *IEEE Spectrum*, pages 34–37, February 1992.
- [Par87] Kenneth Parker. *Integrating Design and Test: Using CAE Tools for ATE Programming*. IEEE Computer Society Press, 1987.
- [Par89] Kenneth P. Parker. The impact of boundary scan on board test. *IEEE Design & Test of Computers*, 6(3):18–30, August 1989.
- [Par92] Kenneth Parker. *The Boundary Scan Handbook*. Kluwer Academic Publishers, 1992.
- [Ray80] Douglas W. Raymond. In-circuit and functional ATE: What's the real difference? In *IEEE Test Conference*, pages 315–317. IEEE Computer Society Press, 1980.
- [Sch89] Rohde & Schwarz. Tester compendium, 1989. Info. No. PD 0756.7565.21.

- [SM90] Richard Sedmak and Colin Maunder. Benefits and penalties of boundary scan. In *Test Access Port and Boundary Scan Architecture*, chapter 12, pages 141–149. IEEE Computer Society Press, 1990.
- [Teg93] Mick Tegethoff. IEEE 1149.1: How to justify implementations. In *IEEE International Test Conference*, page 265. IEEE Computer Society Press, 1993.
- [TL81] Kemon Taschioglou and H. S. Lahman. Functional board testing: the alternatives for increased throughput. *Electronic Packaging and Production*, pages 173–183, June 1981.
- [Wag87] P. Wagner. Interconnect testing with boundary scan. In *IEEE International Test Conference*, pages 52–57. IEEE Computer Society Press, 1987.
- [WE93] Neil Weste and Kamran Eshraghian. *Principles of CMOS VLSI Design — A Systems Perspective*. Addison-Wesley Publishing Company, 2nd edition, 1993.
- [WMR90] T. Winter, M. Muris, and H. Randgaard. Fault diagnosis algorithms for PCBs with boundary scan test. Technical report, ESPRIT Report R7.1a, October 1990. Project 2478 (Research into Boundary Scan Test Implementation).
- [WP83] Thomas W. Williams and Kenneth P. Parker. Design for testability — a survey. *Proceedings of the IEEE*, 71(1):98–112, January 1983.
- [YJ89] Chi W. Yau and Najmi Jarwala. A unified theory for designing optimal test generation and diagnosis algorithms for board interconnects. In *IEEE International Test Conference*, pages 71–77. IEEE Computer Society Press, 1989.



FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



000034578

