



Teste de FPGAs

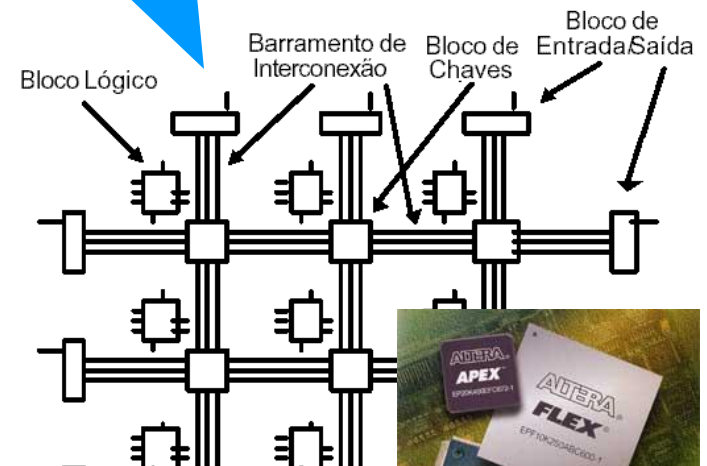
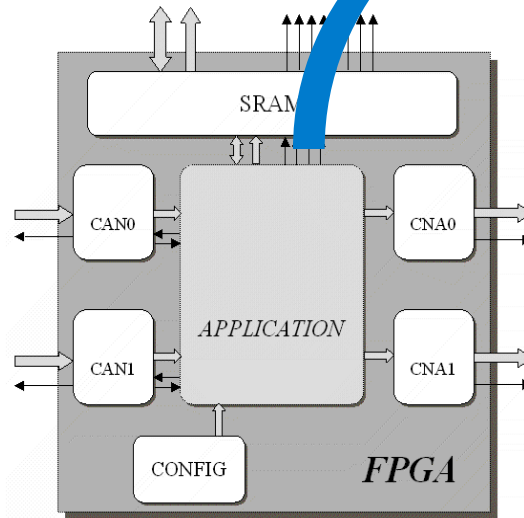
Eric Fabris
Marcelo Lubaszewski



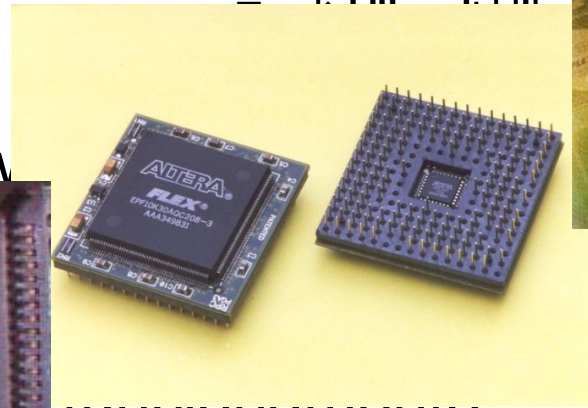
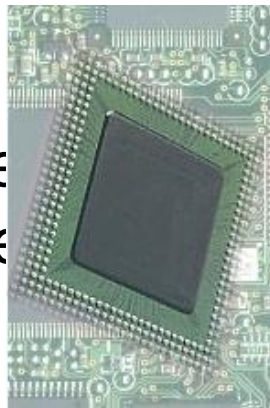
Teste de FPGAs

O que é um FPGA?

FPGA



- Implementa lógica em hardware
- Maior densidade de integração
- Permite reconfiguração
- Permite integração de periféricos



O que é um FPGA?

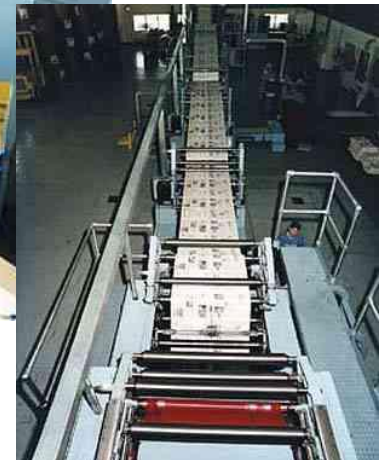


FPGA

- Flexibilidade
- Rápida prototipação
- Uso de tecnologias de ponta
- Baixo custo para baixa demanda

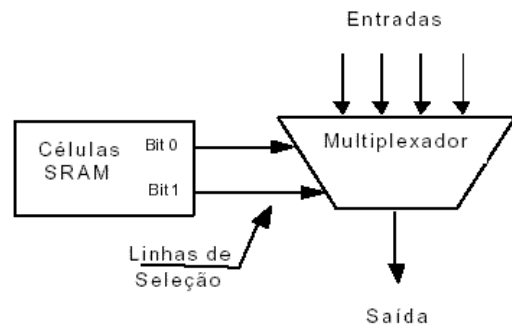
circuito integrado
dedicado

ASIC



Como ligar os elementos programáveis?

SRAM



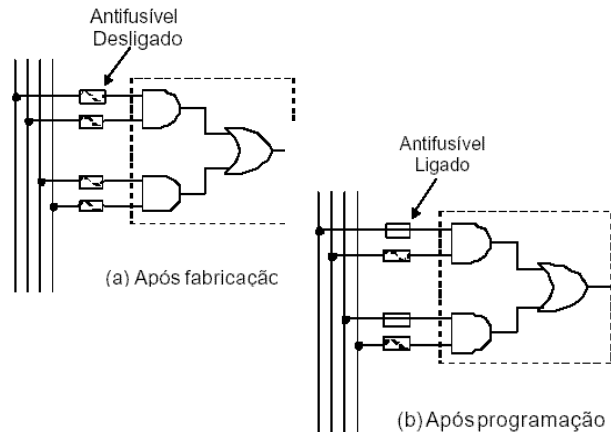
- Exige memória externa
- Necessidade de grande área
- Reconfigurabilidade:
 - rápida
 - ilimitada
- Fabricação simples

XILINX®

ALTERA

{algotronix}

Antifusível

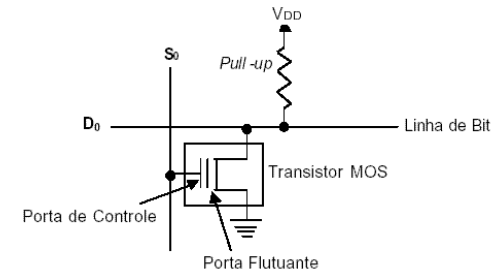


- Circuito para queimar lógica
- Pequeno tamanho
- R e C baixos

QUICKLOGIC®
Embedded Standard Products... Beyond Programmable Logic

Actel

Porta Flutuante



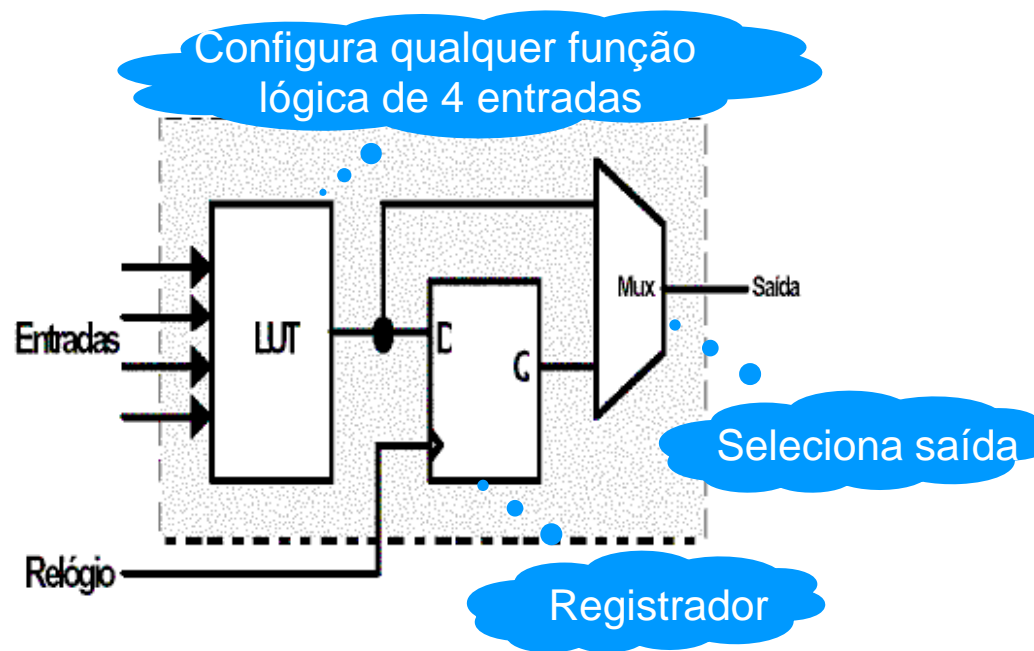
- Similar a EPROM
- Remoção de lógica com UV
- Reconfigurabilidade
- R e C maiores
- Fabricação mais complexa

AMD

Lattice®

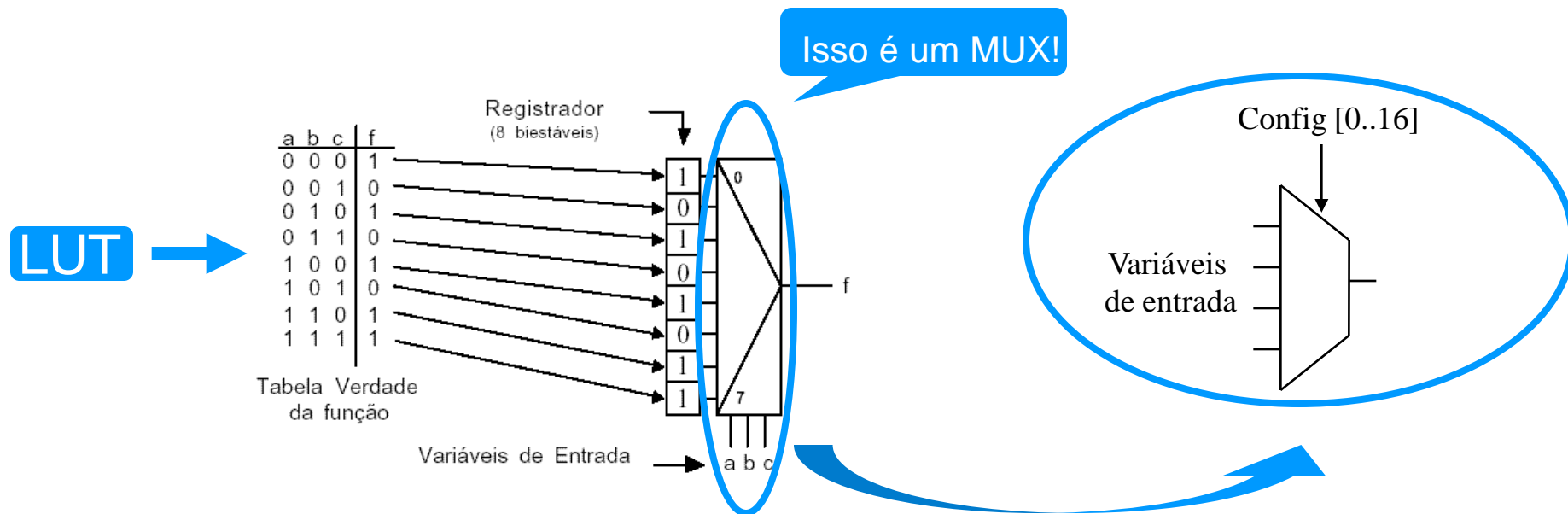
Blocos Lógicos (1)

- Comumente chamados de CLBs
- Compostos basicamente por 3 sub-blocos
 - LUT (look-up table)
 - Registrador
 - Multiplexador

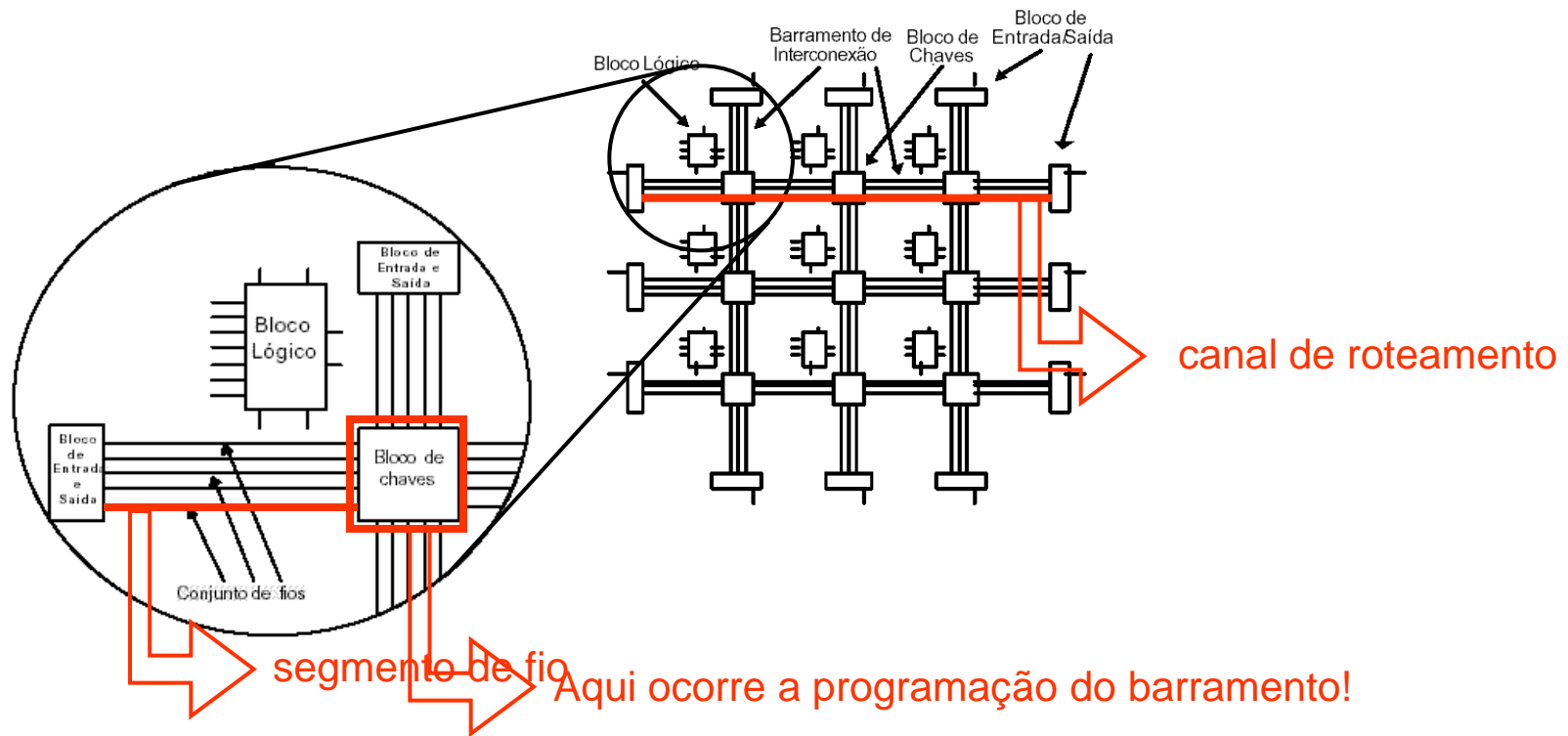


Blocos Lógicos (2): LUT

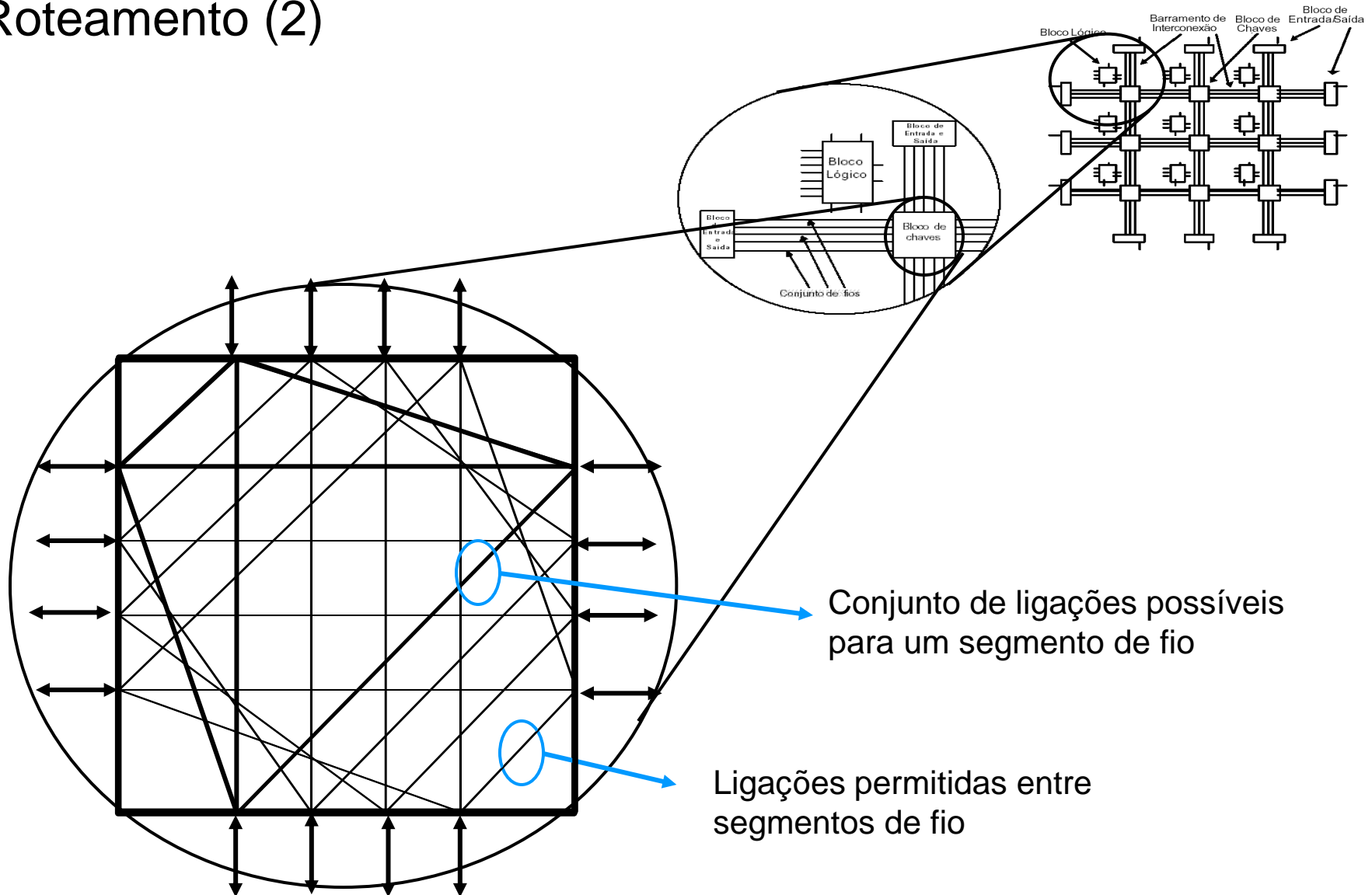
- Configura qualquer função lógica de 4 entradas
- Pode ser vista como um multiplexador
 - Endereços são os dados da programação (que porta é esta?)
 - Seleção é o sinal de entrada
- Alto custo de área para implementação
- Grande flexibilidade



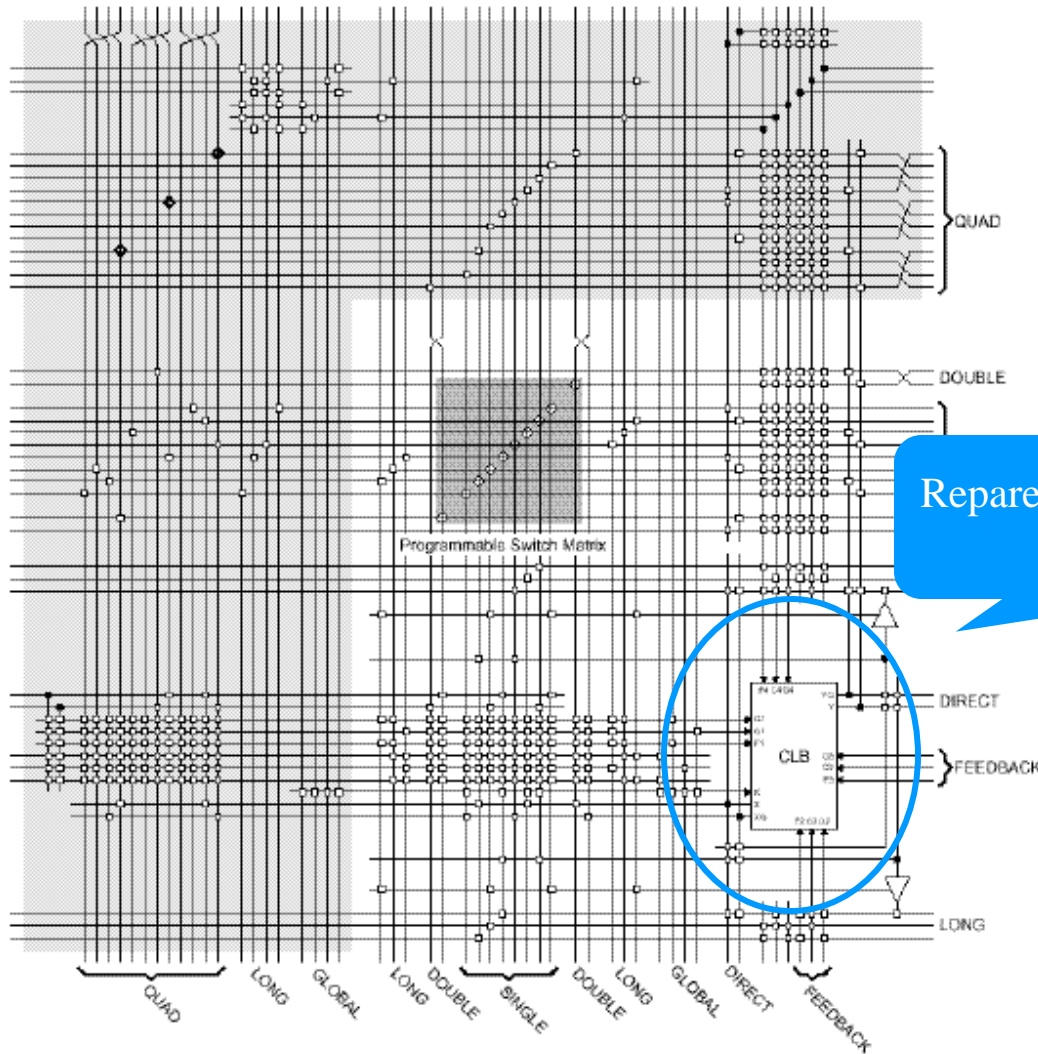
Roteamento (1)



Roteamento (2)



Roteamento (3)

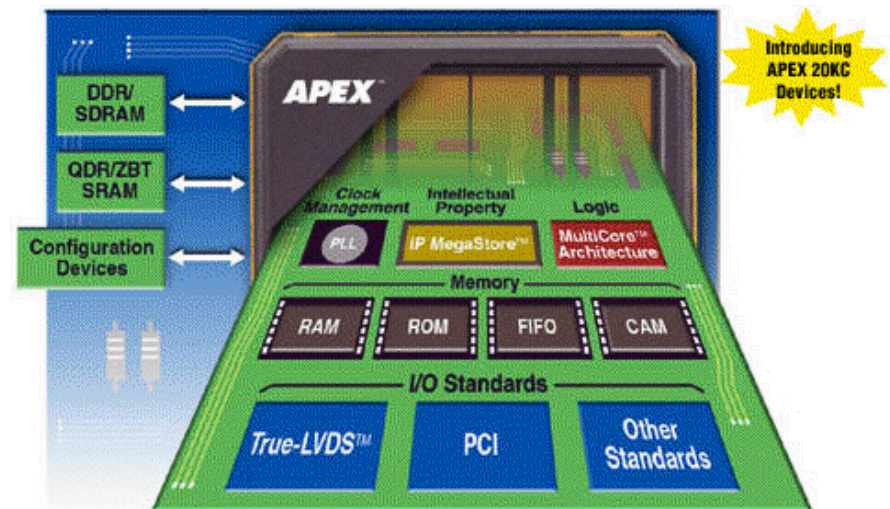


- Roteamento consome de 70 a 90% da área total!
- Segmentos de metal são unidos através de chaves programáveis
- Chaves programáveis implementadas com transistores de passagem
- Reduz performance do dispositivo

Repare na área do CLB comparada com o roteamento!

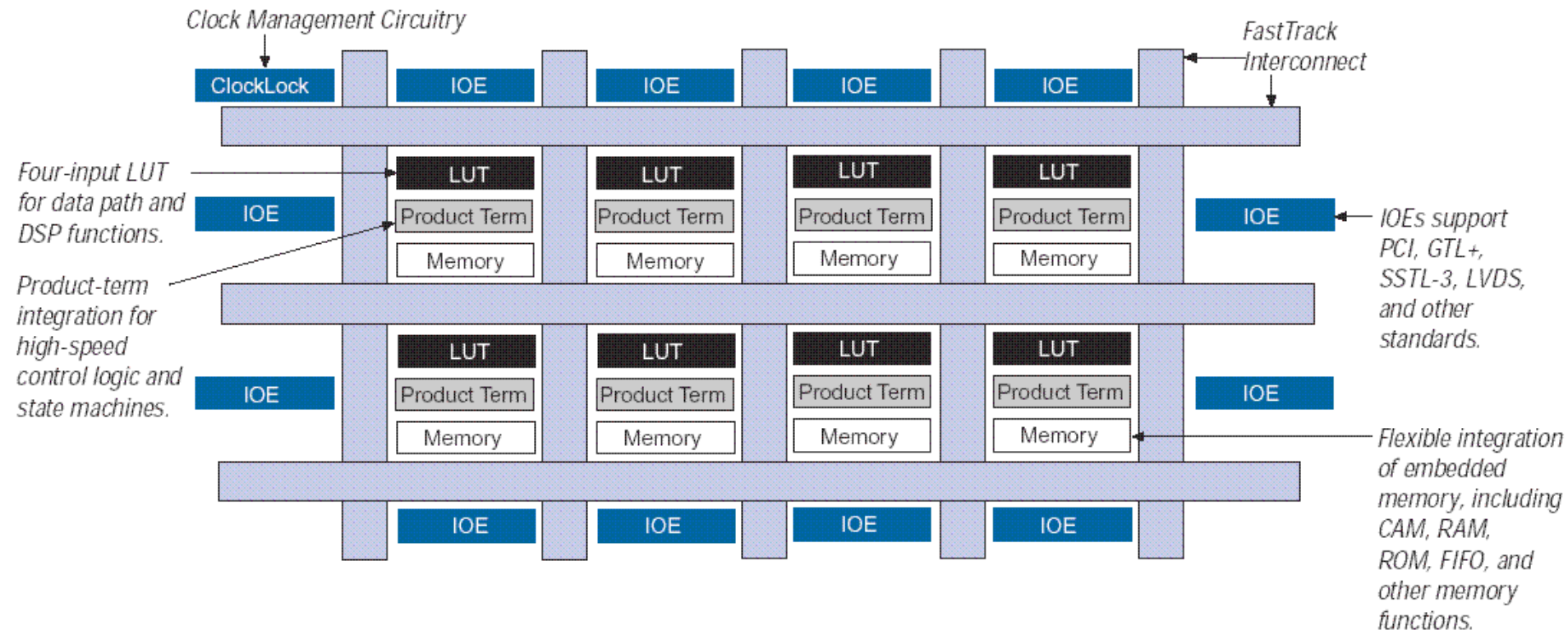
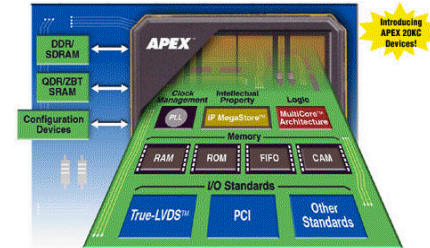
Novos FPGAs

- Usam conjunto de blocos lógicos básicos
 - permitem redes locais de roteamento
 - alto desempenho
- Apresentam módulos com funcionalidade específica
 - memórias acesso simples e duplo, PLLs, multiplicadores, etc
 - maior eficiência em tarefas comuns
- Capacidades especiais
 - Reconfigurabilidade parcial (Xilinx)
- Agregação de IP Hard e Soft Cores



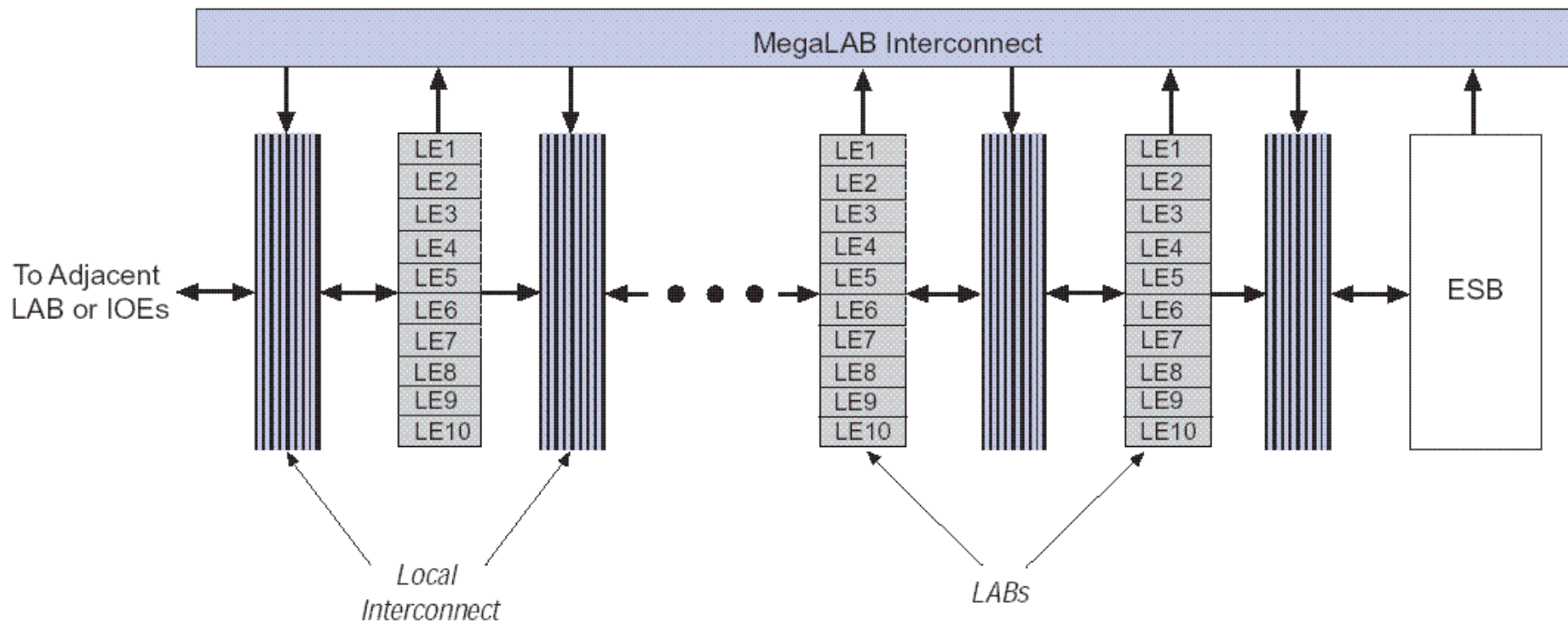
Altera Apex 20K (1)

- Interconexões locais
- Memória embutida distribuída pelo circuito
- Não possui reconfigurabilidade parcial



Altera Apex 20K (2)

- Estrutura locais (MegaLAB)
 - 10, 16 ou 24 LUTs
 - alto desempenho
- Menor uso do roteamento global



Métodos de Teste da Manufatura

Teste Funcional

VS

Teste Estrutural

- não otimizado
- não automatizado
- sem parâmetros para aplicação do teste

- sem modelo de falhas
- sem ferramentas

Métodos de Teste da Manufatura

Teste Estrutural do FPGA:

- . Teste das Células de Base**
- . Teste das Interconexões**
- . Teste da Memória**

Teste das Células de Base (1)

Para cada possível bloco é necessário:

- Gerar um modelo de falhas
- Gerar configurações de teste e vetores de teste
- Minimizar o número de configurações de teste

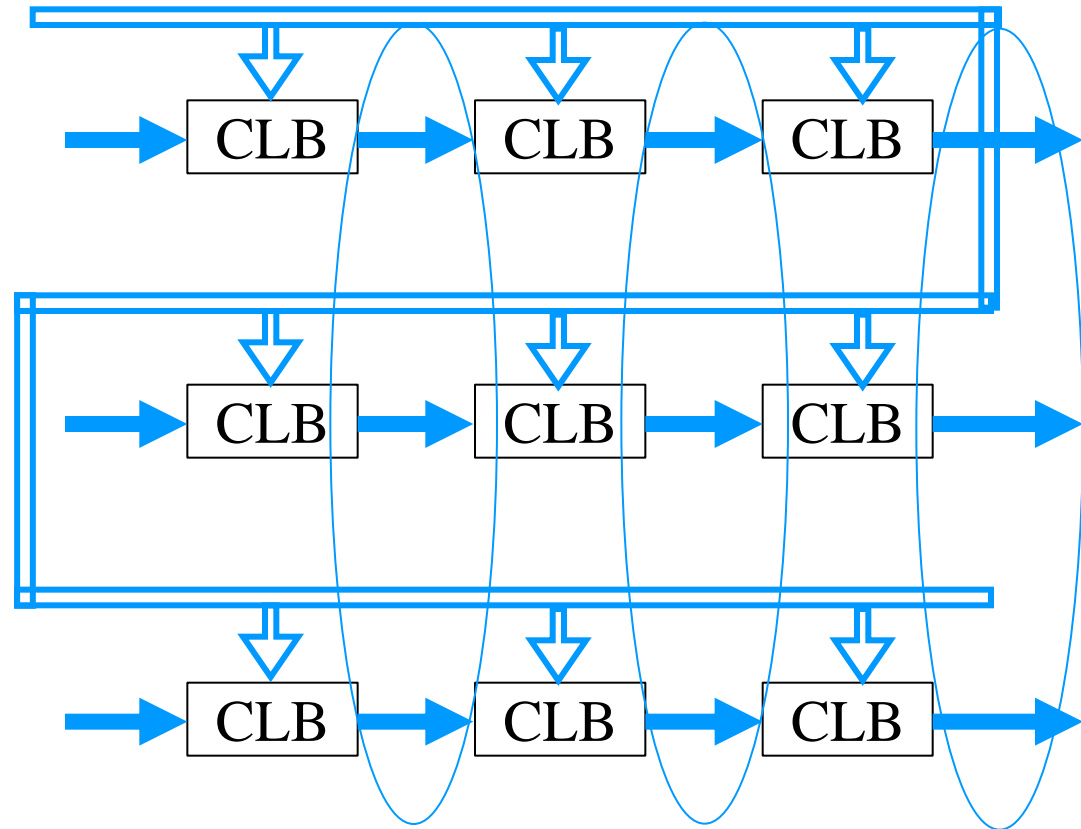
Número de configurações necessárias possui grande influência no custo do teste.

Teste das Células de Base (2)

Objetivo : Testar várias células de base simultaneamente.

Problemas :

- ✗ Gerar a seqüência de teste para células que não podem ser acessadas diretamente por entradas primárias.
- ✗ Observar a saída dos CLBs nas saídas primárias do circuito.

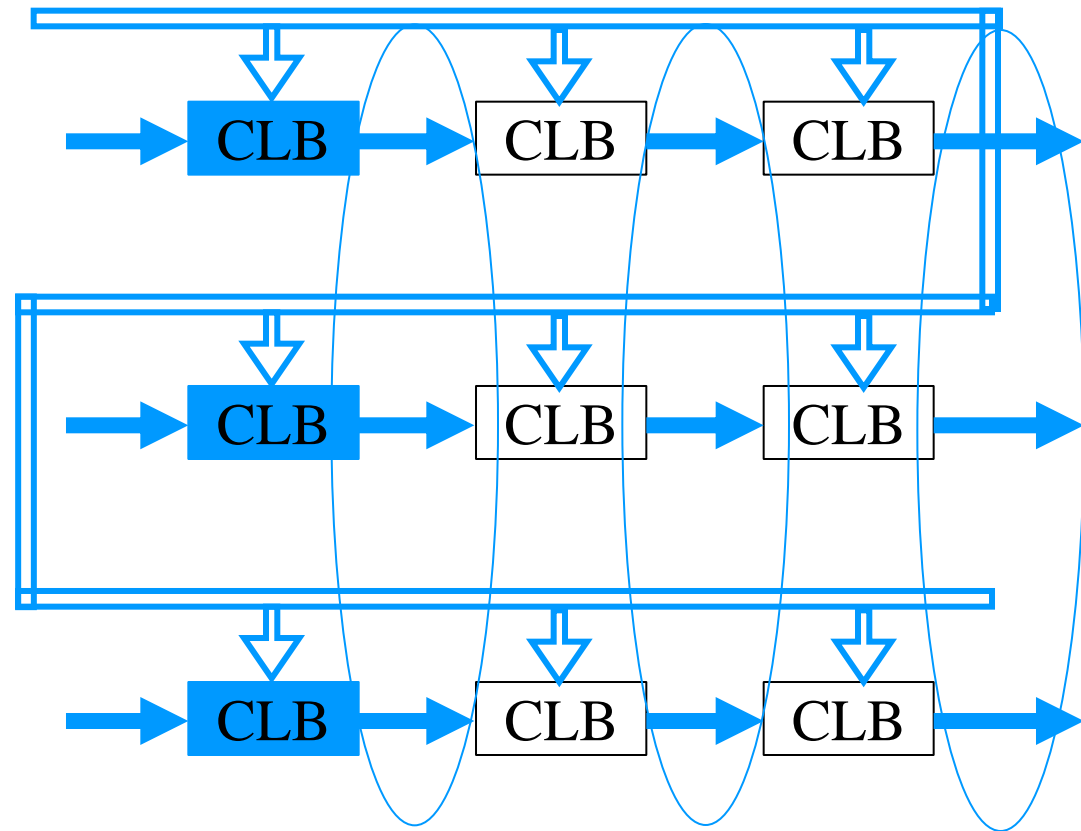


Teste das Células de Base (2)

Objetivo : Testar várias células de base simultaneamente.

Problemas :

- ✕ Gerar a seqüência de teste para células que não podem ser acessadas diretamente por entradas primárias.
- ✕ Observar a saída dos CLBs nas saídas primárias do circuito.

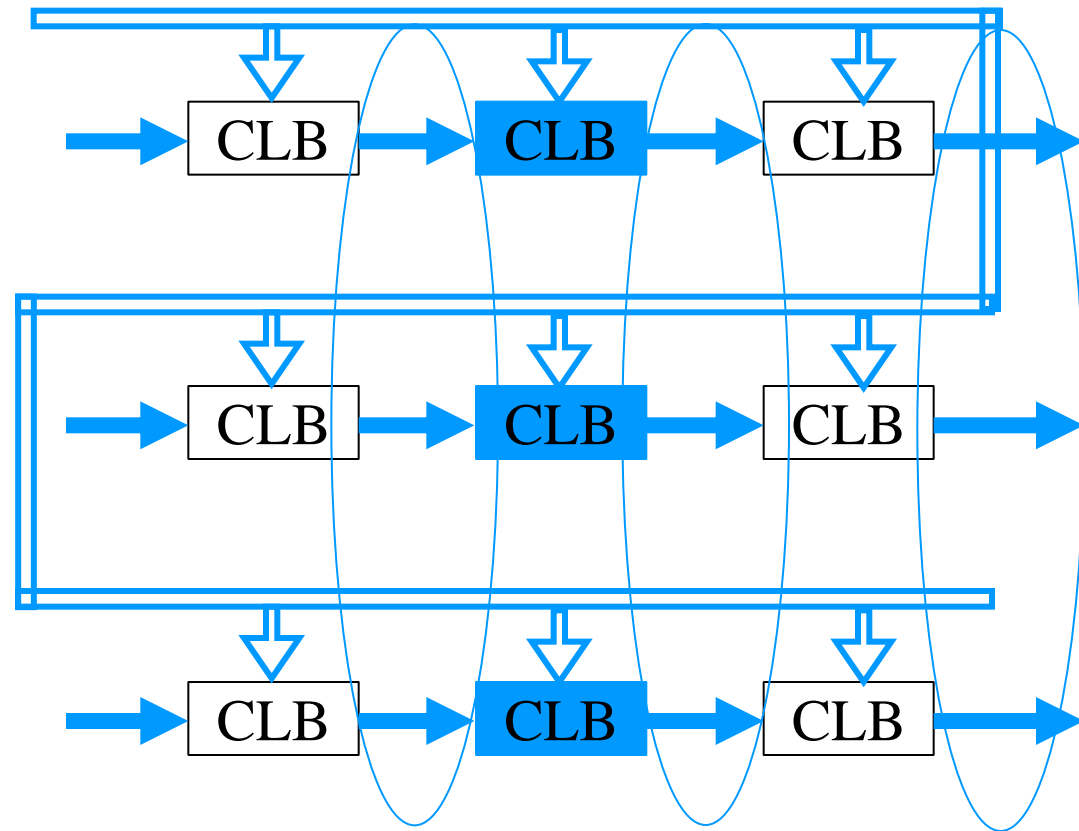


Teste das Células de Base (2)

Objetivo : Testar várias células de base simultaneamente.

Problemas :

- ✗ Gerar a seqüência de teste para células que não podem ser acessadas diretamente por entradas primárias.
- ✗ Observar a saída dos CLBs nas saídas primárias do circuito.

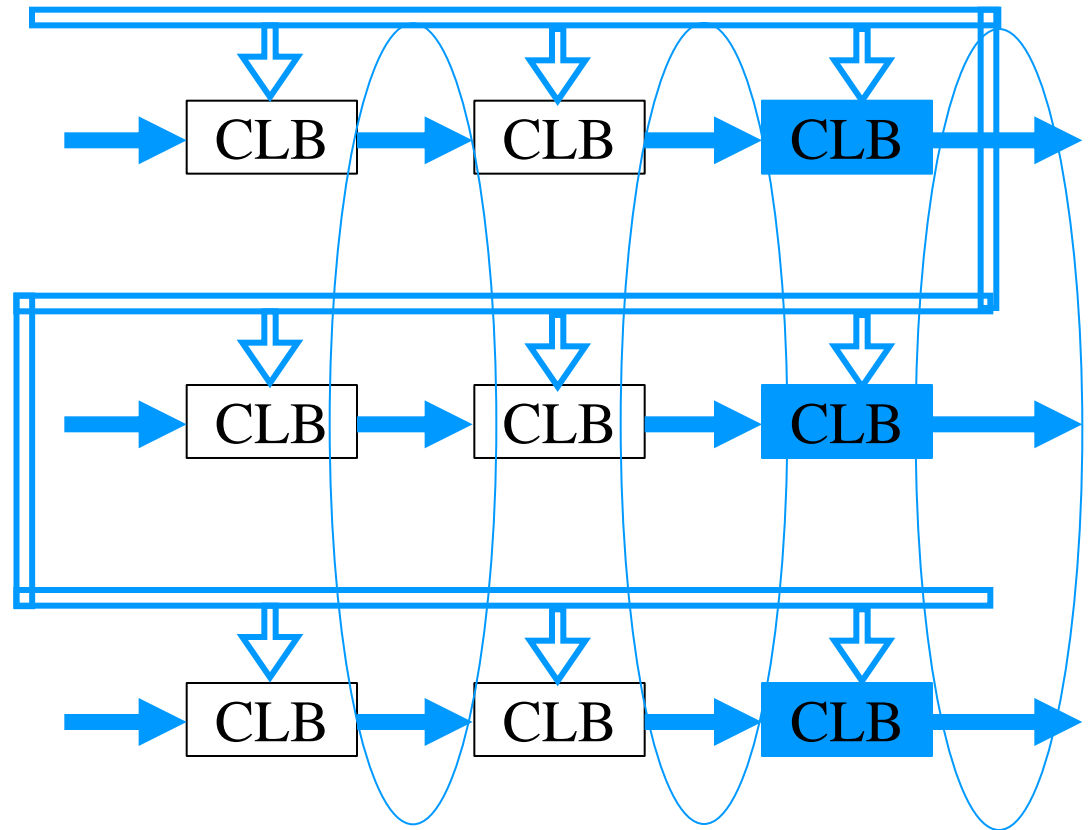


Teste das Células de Base (2)

Objetivo : Testar várias células de base simultaneamente.

Problemas :

- ✗ Gerar a seqüência de teste para células que não podem ser acessadas diretamente por entradas primárias.
- ✗ Observar a saída dos CLBs nas saídas primárias do circuito.

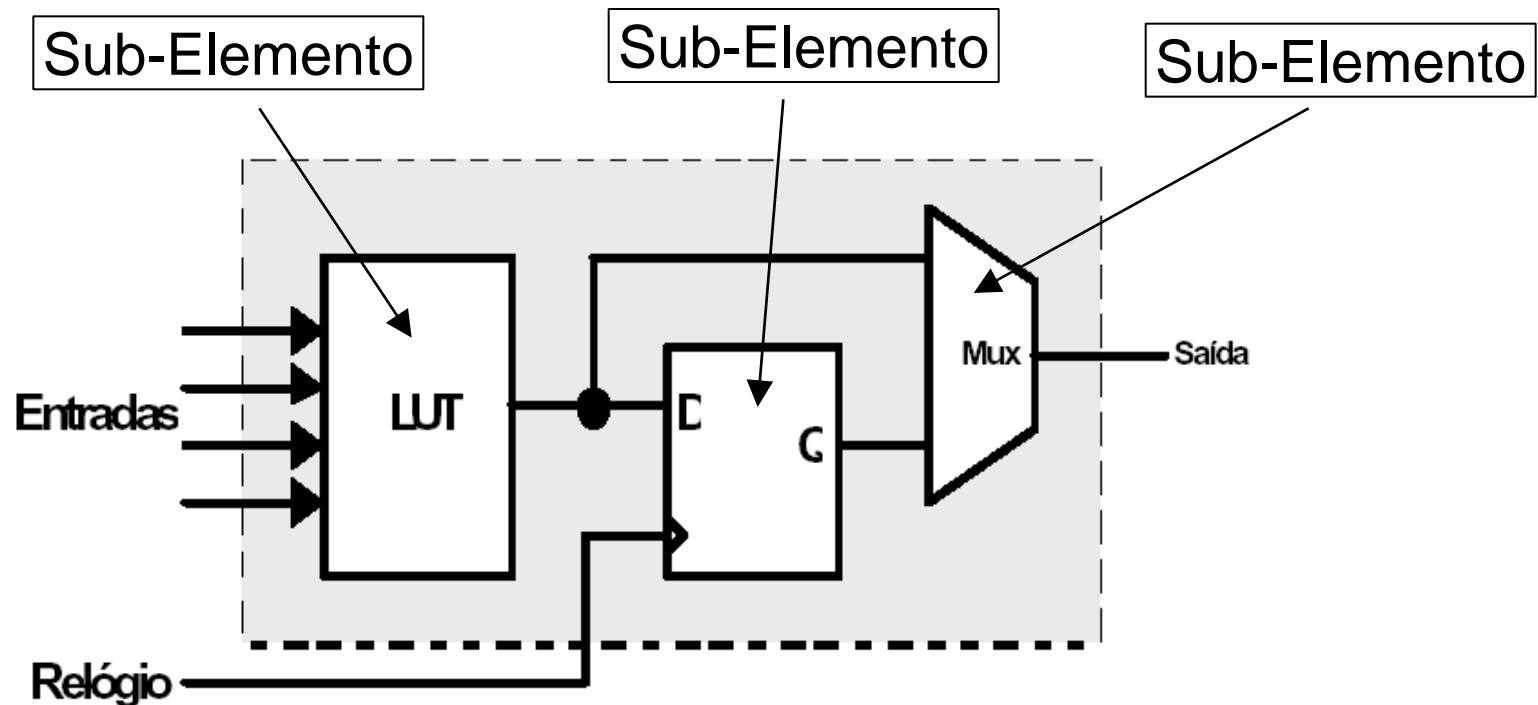


Teste das Células de Base (3)

Como testar uma célula de base?

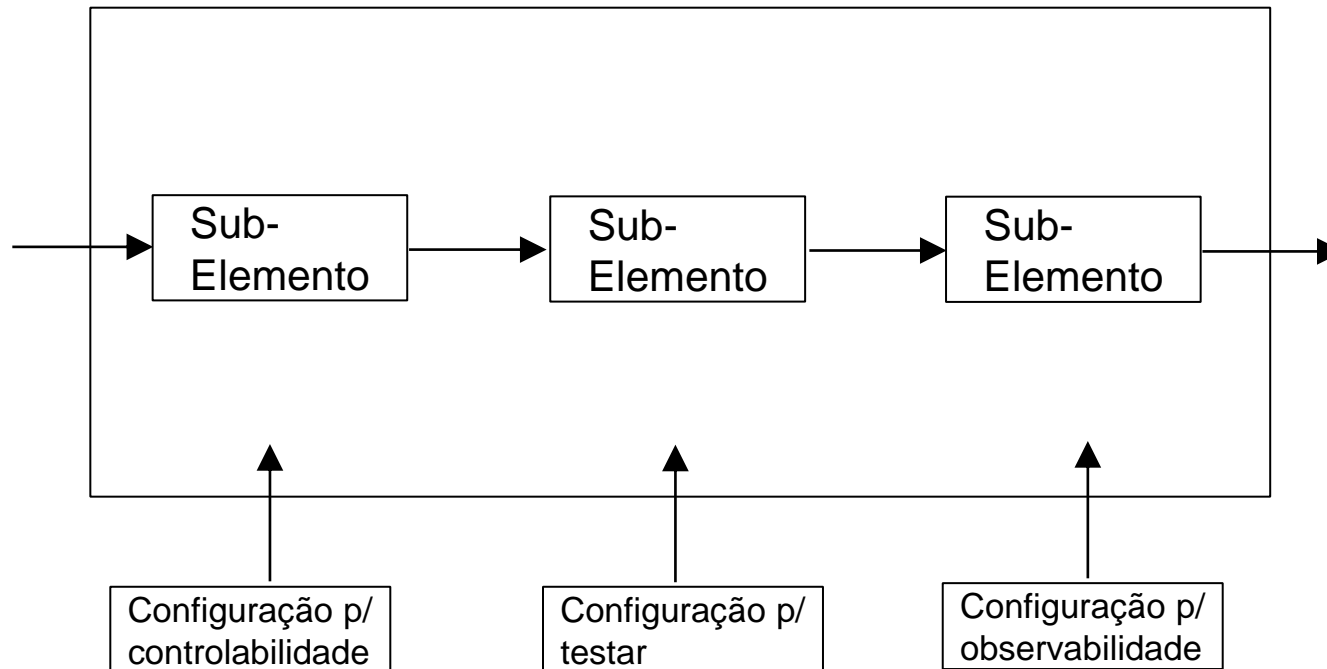
↪ Células de base são divididas em sub-elementos.

↪ Para testar a CLB basta testar os sub-elementos.



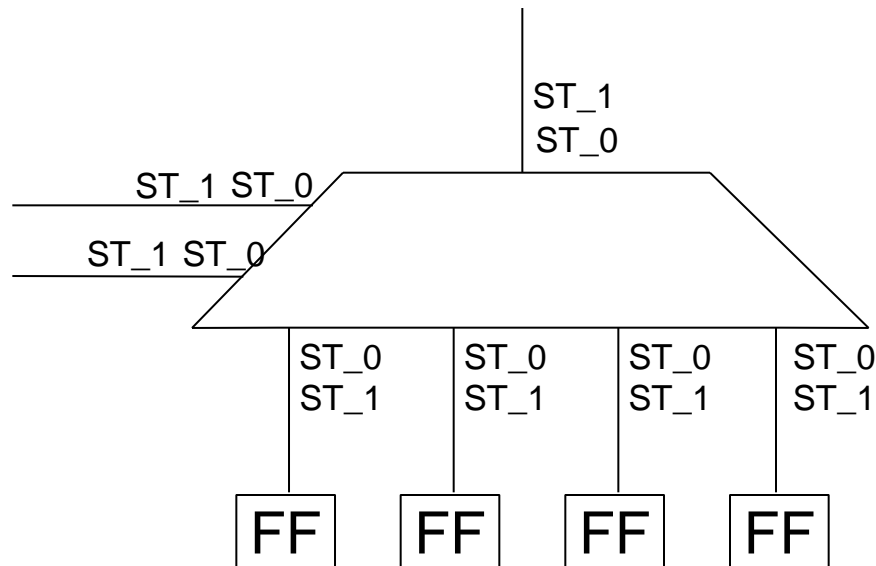
Teste das Células de Base (4)

Como testar os sub-elementos ?



Testando a Look-up Table (1)

Modelo de falhas stuck-at

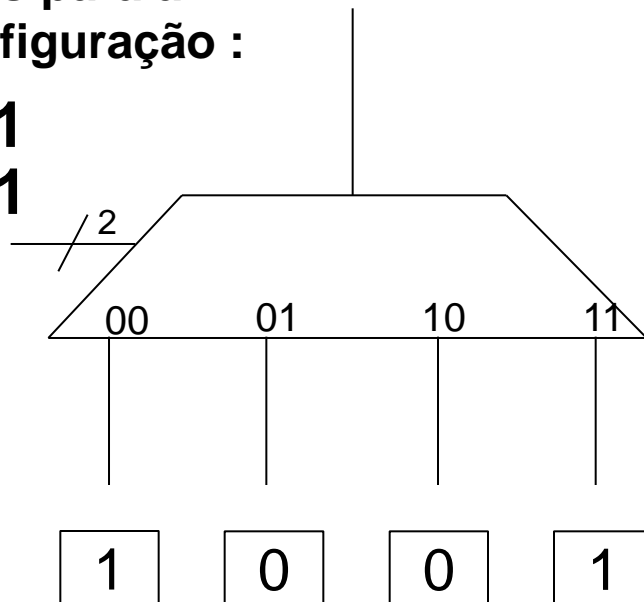


Testando a Look-up Table (2)

Teste com duas configurações

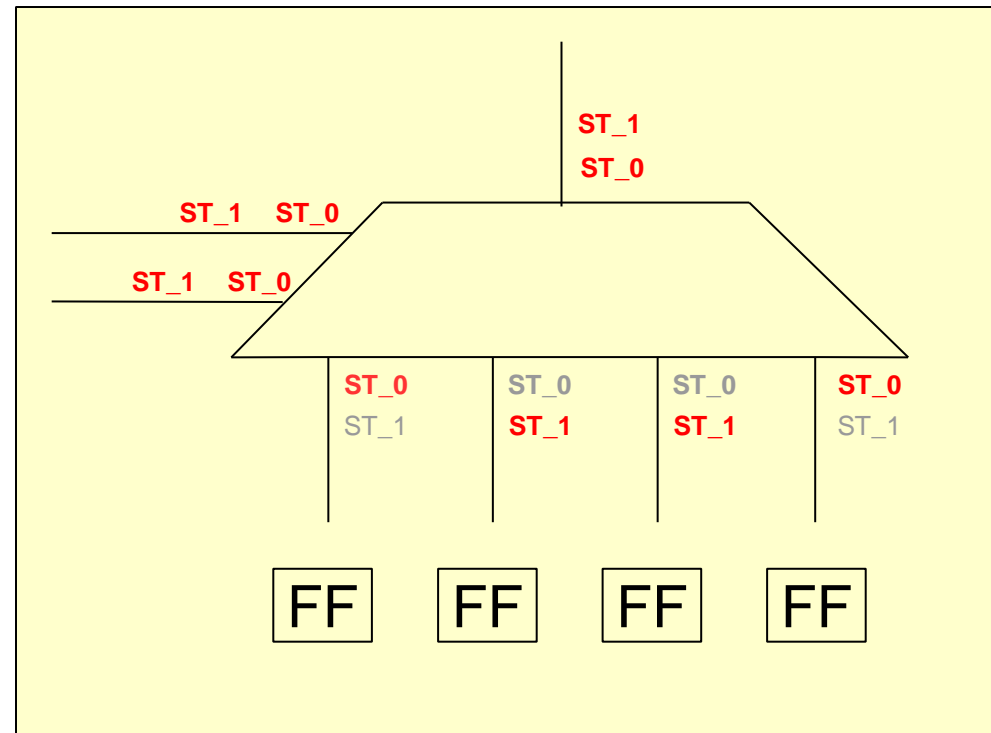
Sequência de
vetores para a
1ª configuração :

0011
0101



1ª Configuração

Falhas da lista de falhas detectadas :

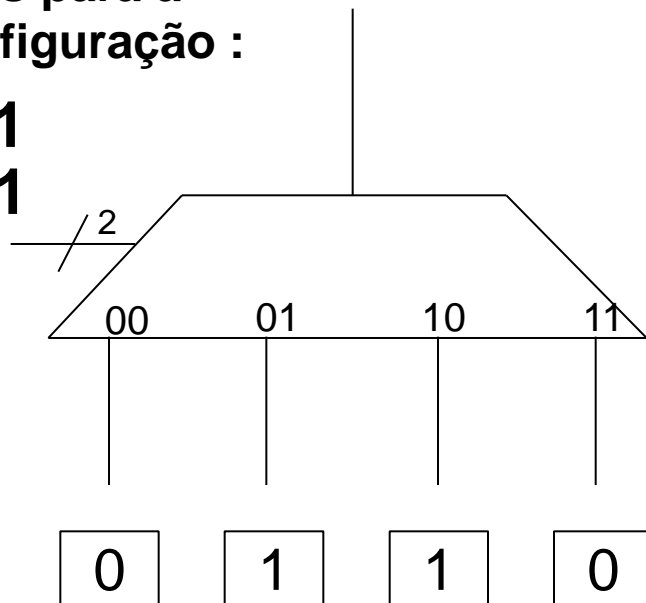


Testando a Look-up Table (3)

Teste com duas configurações

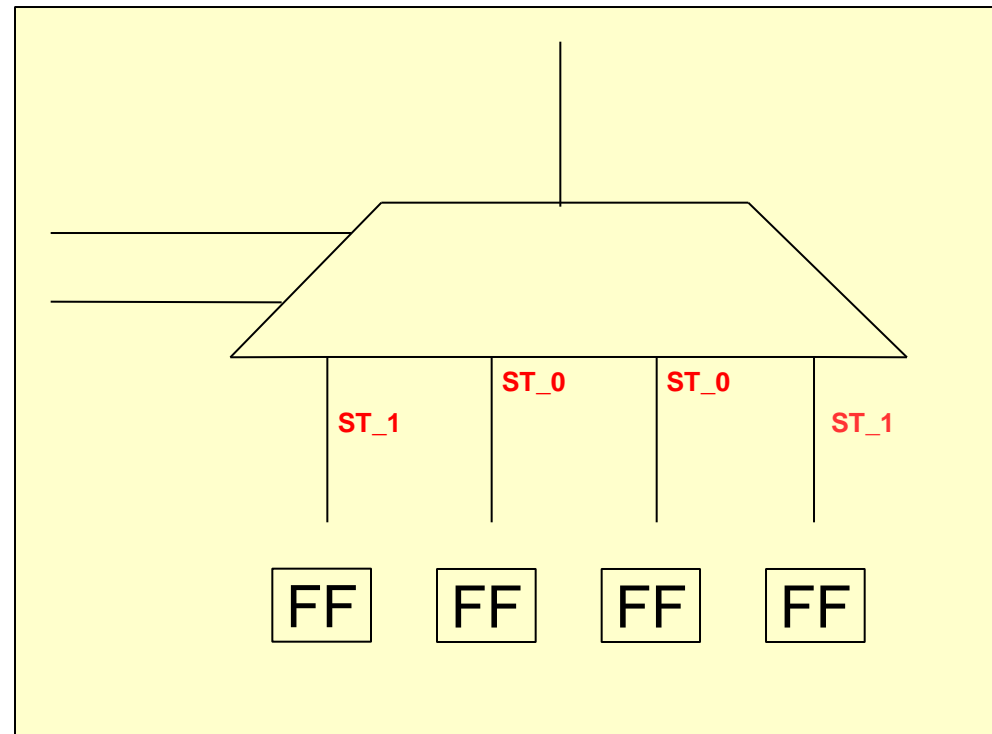
Sequência de
vetores para a
2ª configuração :

0011
0101



2ª Configuração

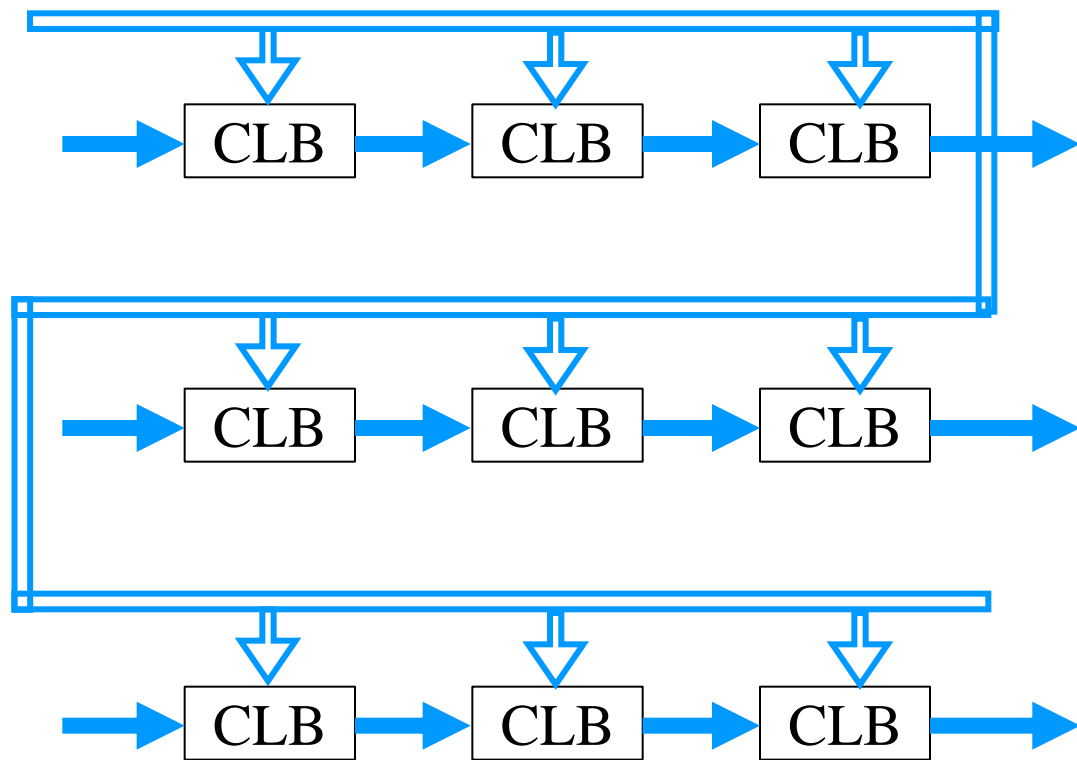
Falhas da lista de falhas detectadas :



Testando a Look-up Table (4)

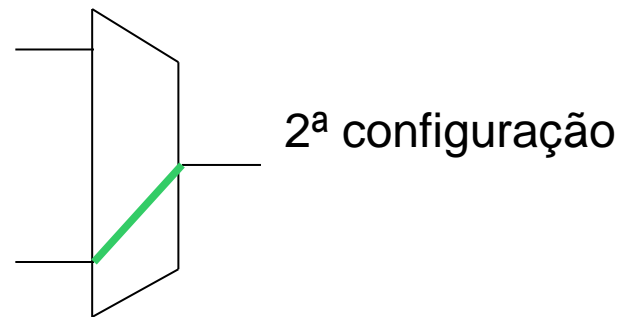
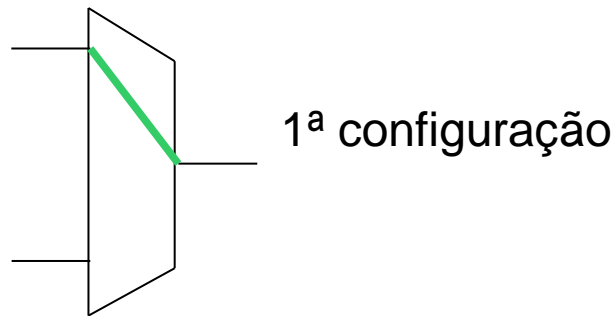
Para identificar todas as falhas de colagem em uma LUT basta configurá-la com duas funções complementares.

As LUTs são configuradas com a função XOR e NXOR para aumentar a controlabilidade e a observabilidade dos CLBs.

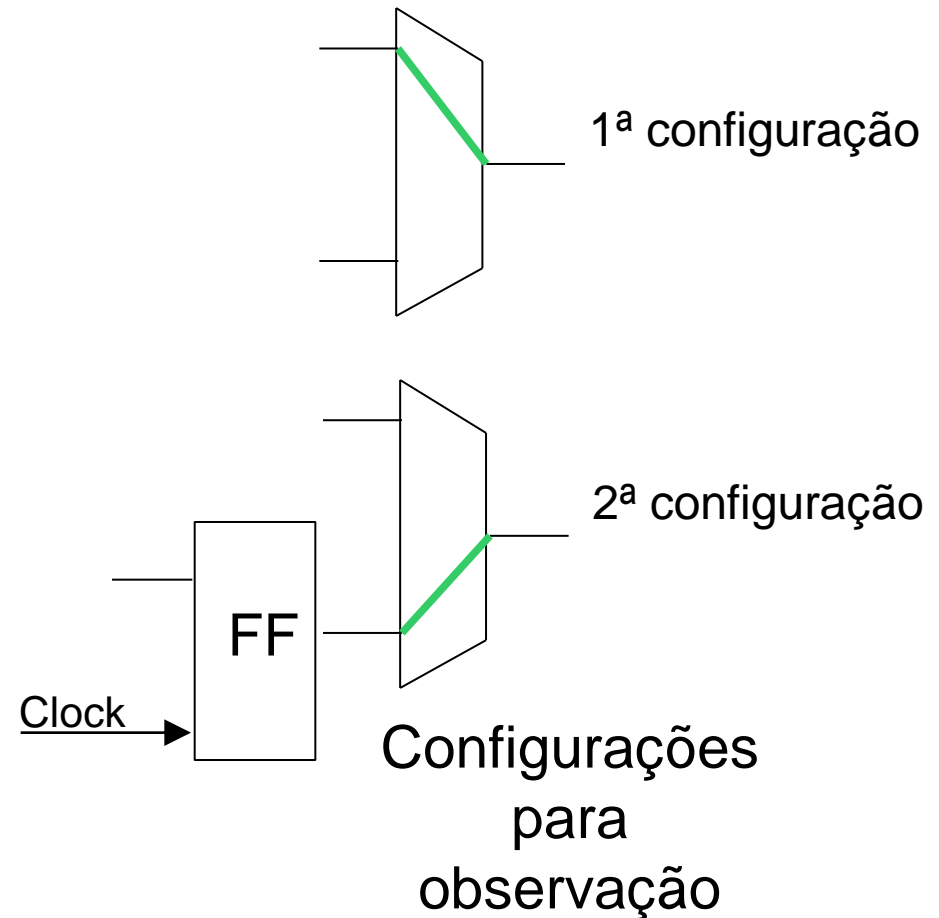


Testando o MUX (1)

O multiplexador: (modelo de falhas stuck-at)



Configurações
para teste

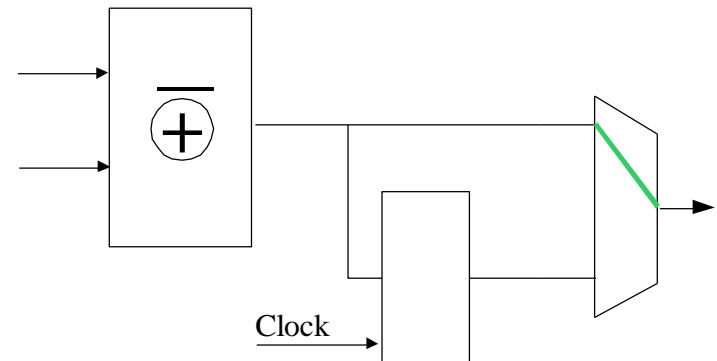
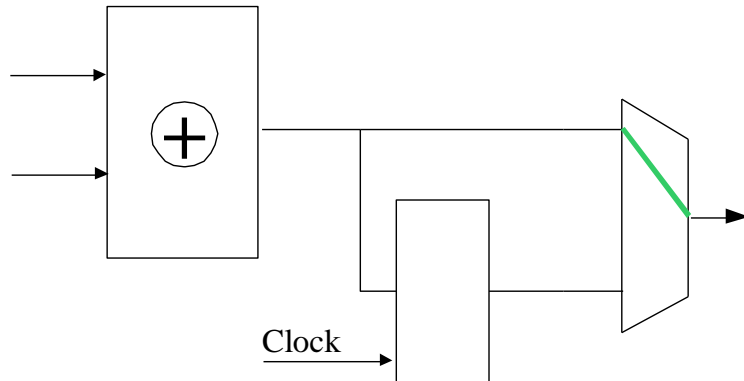


Testando o MUX (2)

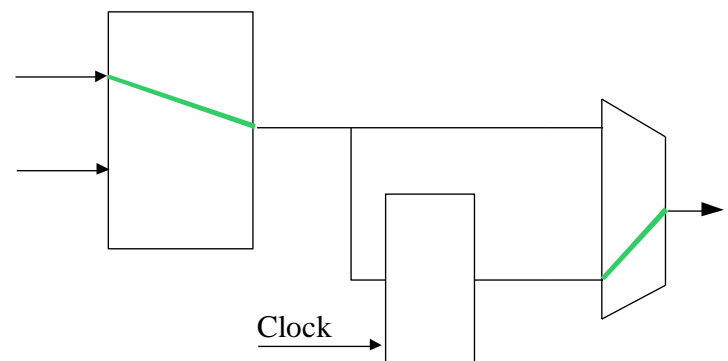
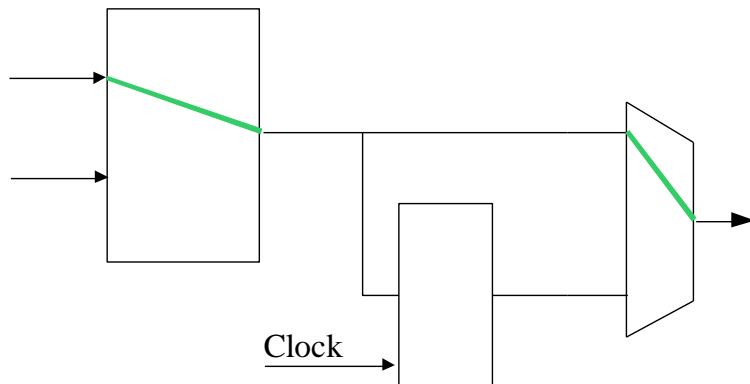
O numero de configurações necessárias é igual ao numero de entradas do mesmo.

Permite observar qualquer sinal de entrada. O sinal que será observado é determinado pela configuração do multiplexador.

Aplicando a todo CLB (1)



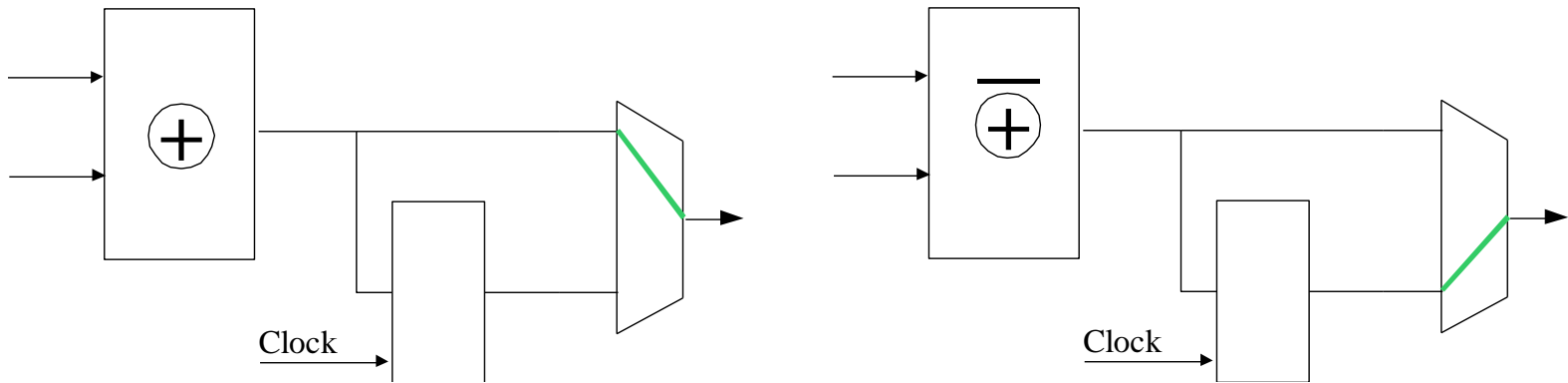
Configurações do CLB para teste da LUT



Configurações do CLB para teste do MUX e FF

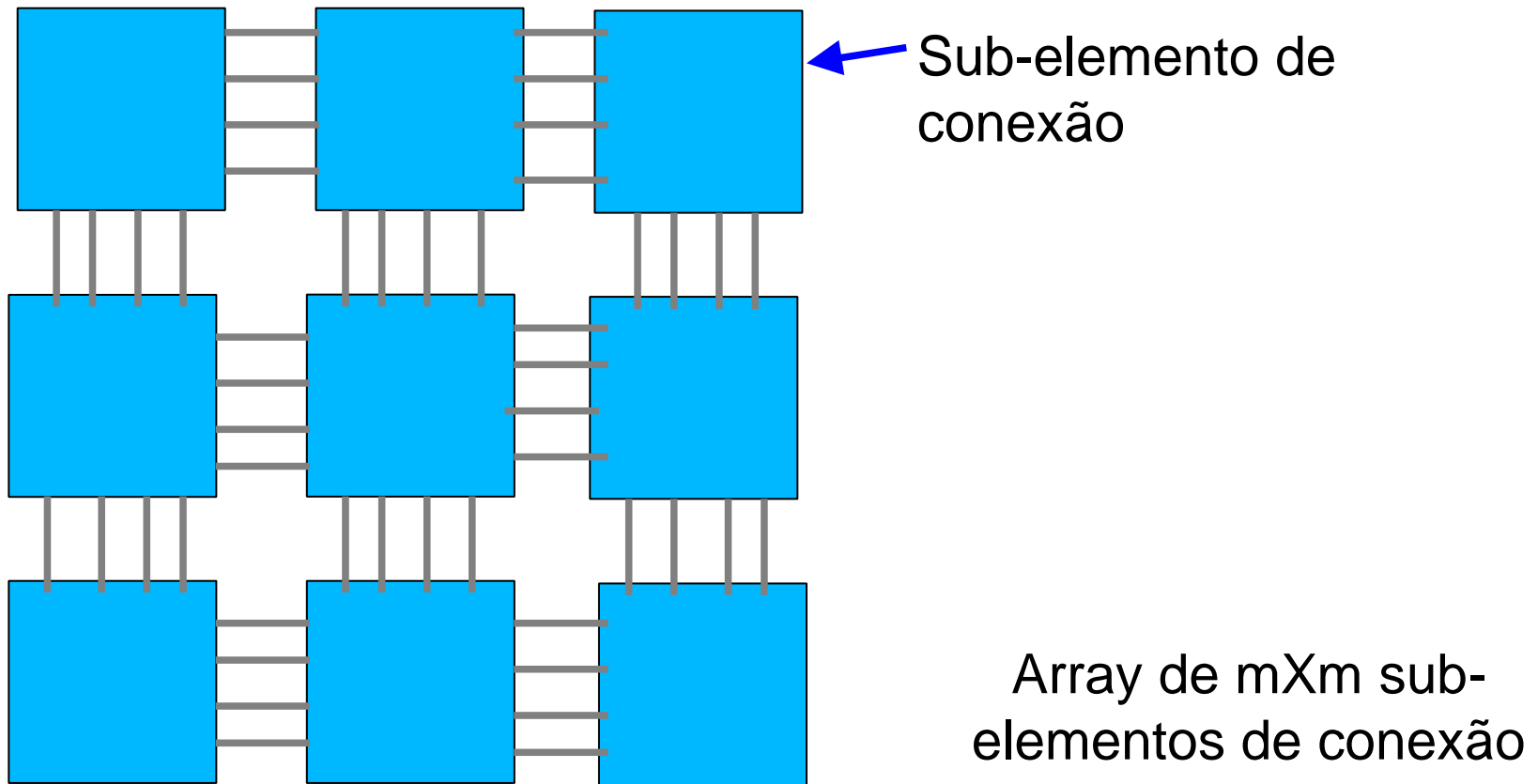
Aplicando a todo CLB (2)

O número de configurações pode ser reduzido.



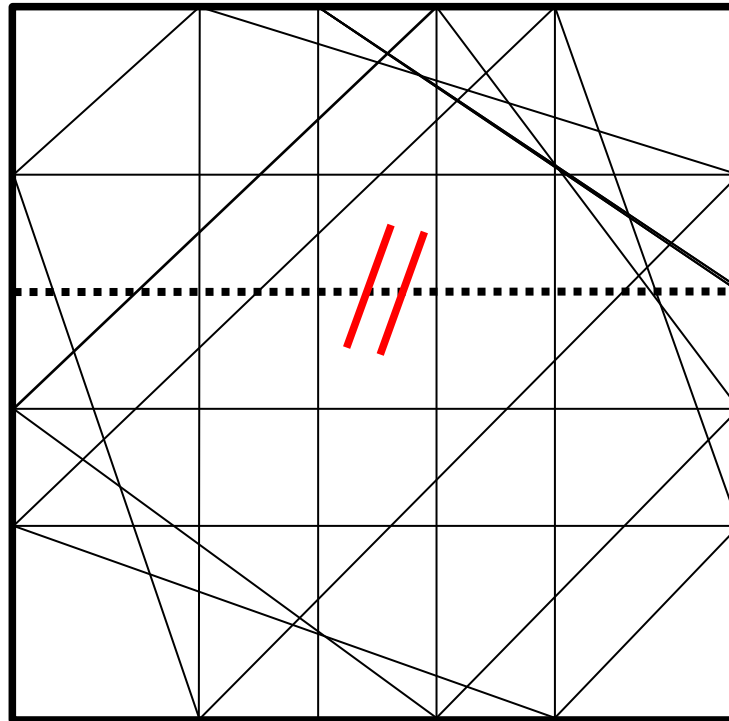
Configurações mínimas necessárias para teste da CLB.

Teste das Interconexões



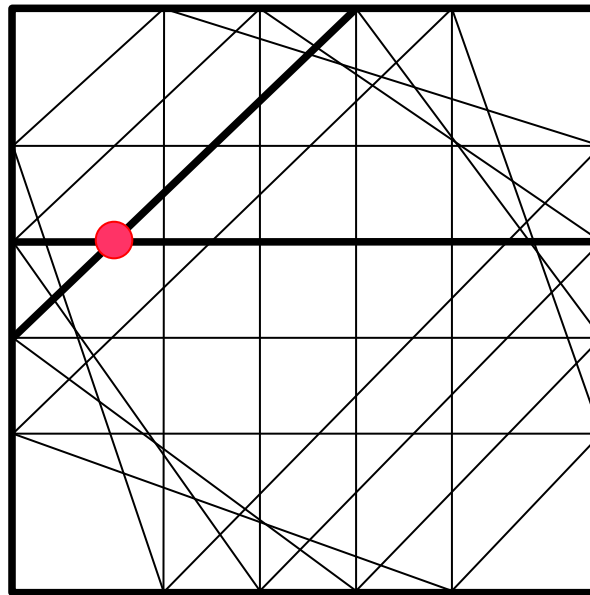
Modelos de Falhas para Teste de Interconexões (1)

Line Open



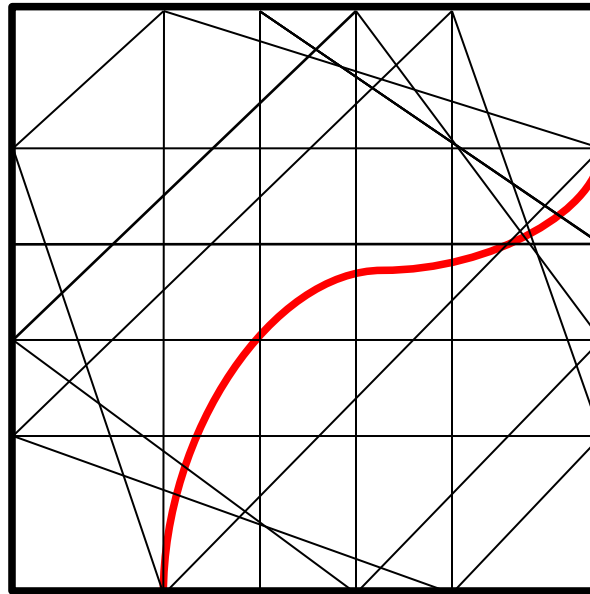
Modelos de Falhas para Teste de Interconexões (2)

LinePair Short



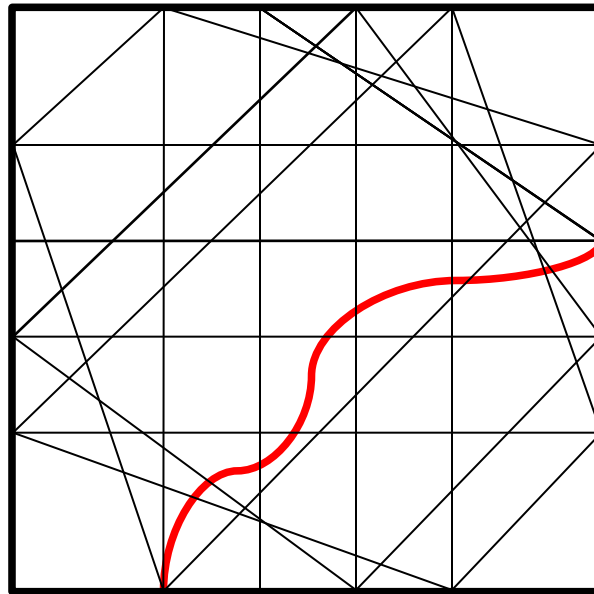
Modelos de Falhas para Teste de Interconexões (3)

Permanentemente Conectados (pinos conectáveis)



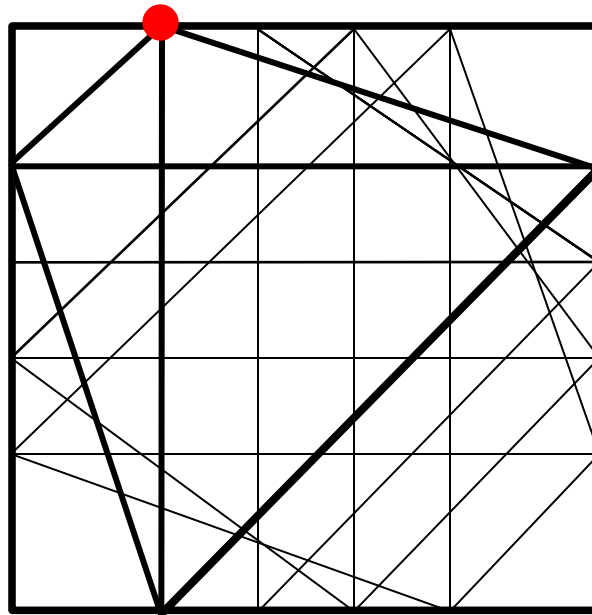
Modelos de Falhas para Teste de Interconexões (4)

Permanentemente Conectados (pinos não conectáveis)



Modelos de Falhas para Teste de Interconexões (5)

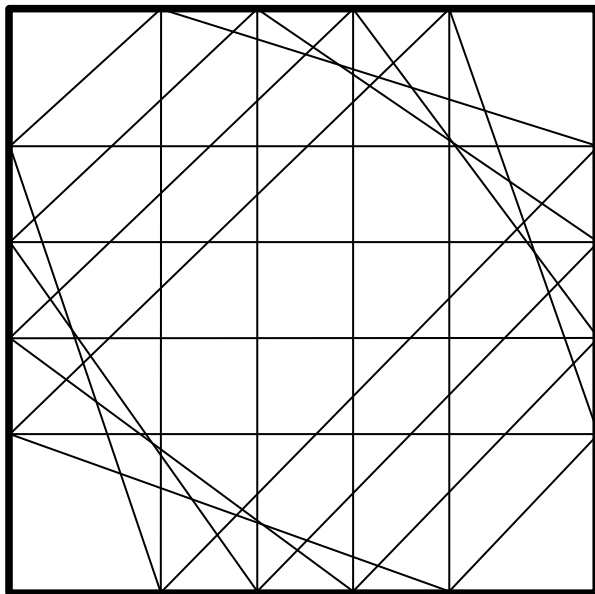
Permanentemente Desconectados (pinos conectáveis)



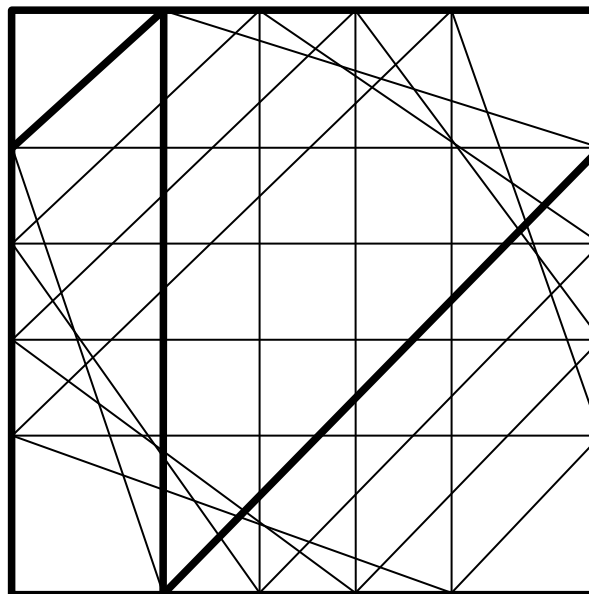
Detecção de falhas nas interconexões (1)

1ª proposta:

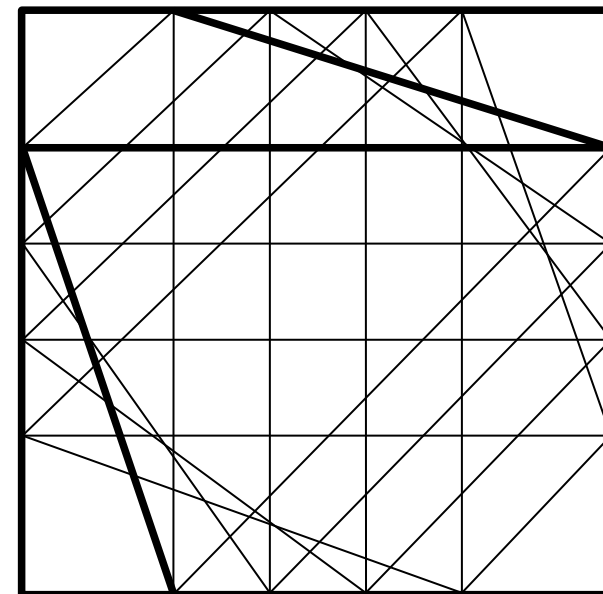
3 Configurações



Configuração sem
conexões



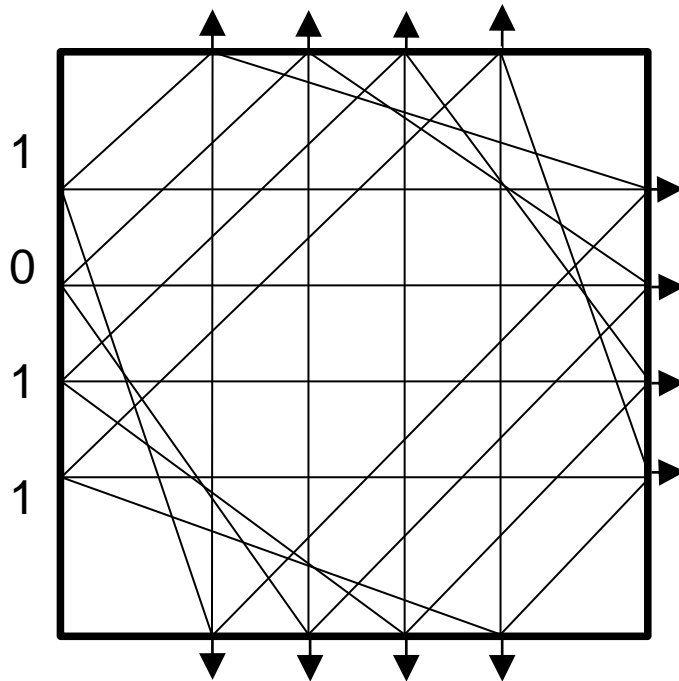
Configuração em N



Configuração em Z

Detecção de falhas nas interconexões (1)

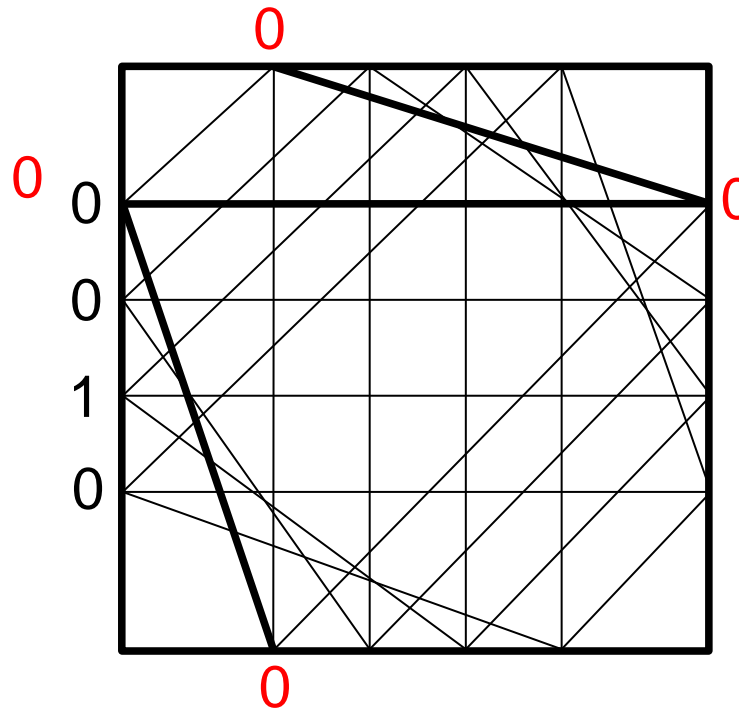
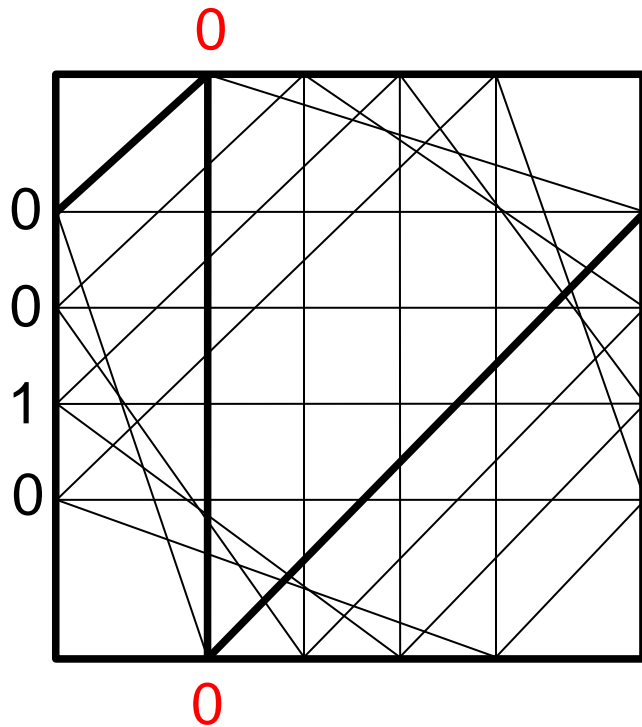
Detecção de permanentemente conectados:



Se algum pino tiver valor lógico igual aos valores do vetor de entrada o pino está permanentemente conectado.

Detecção de falhas nas interconexões (1)

Detecção de linhas abertas e pinos permanentemente desconectados:

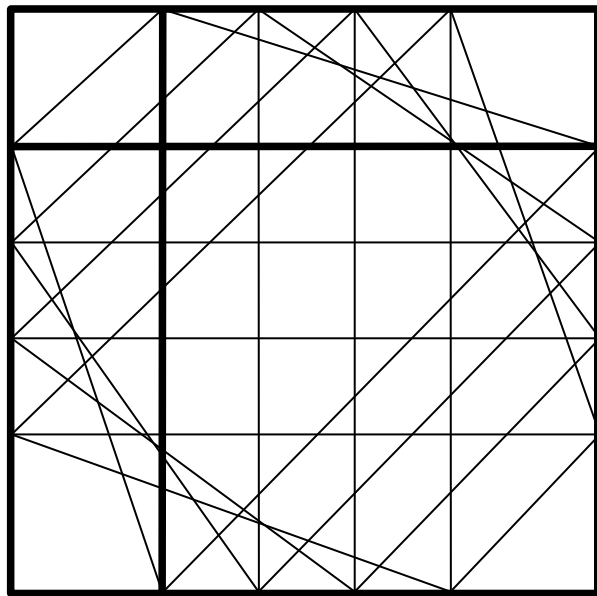


Se algum pino estiver com o valor diferente do valor de entrada existe uma falha na conexão ou no pino.

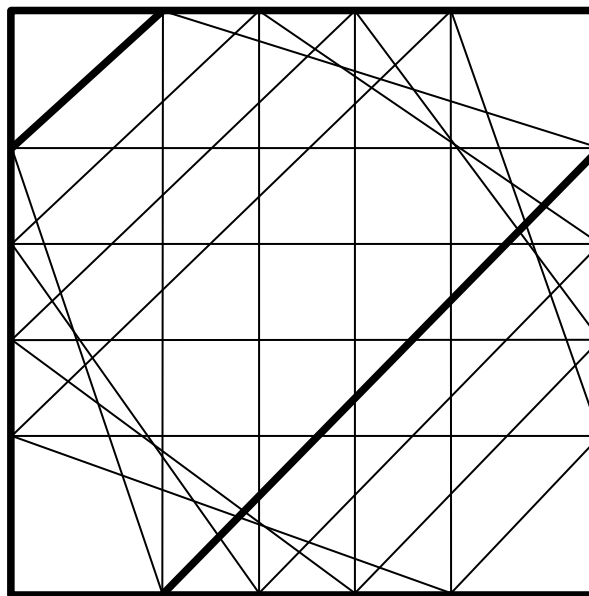
Detecção de falhas nas interconexões (1)

2ª proposta:

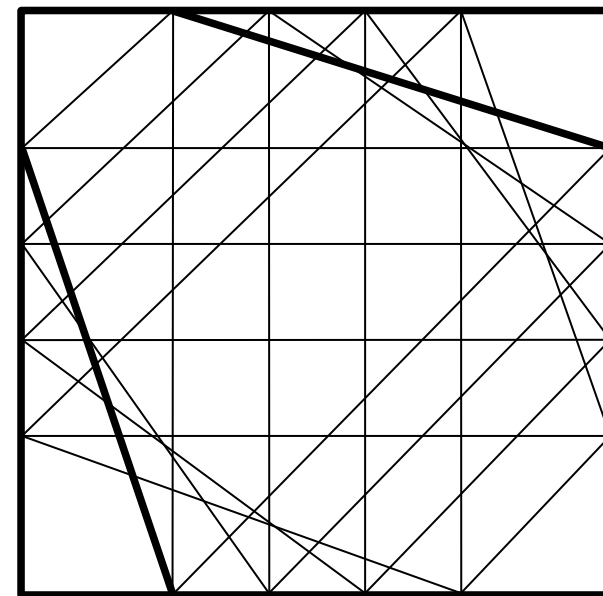
3 Configurações



Configuração
Ortogonal



Configuração
Diagonal 1

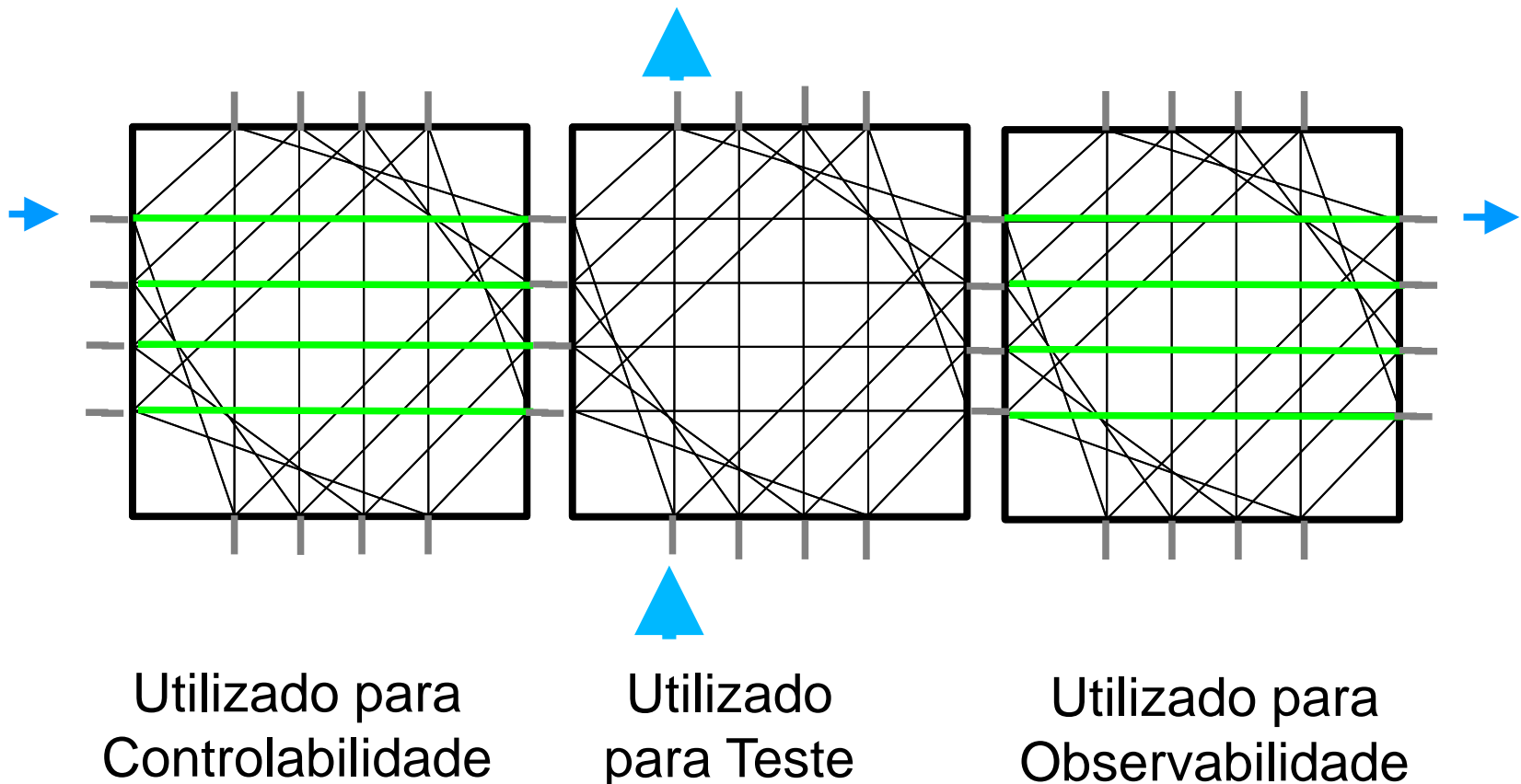


Configuração
Diagonal 2

Detecção de falhas nas interconexões (1)

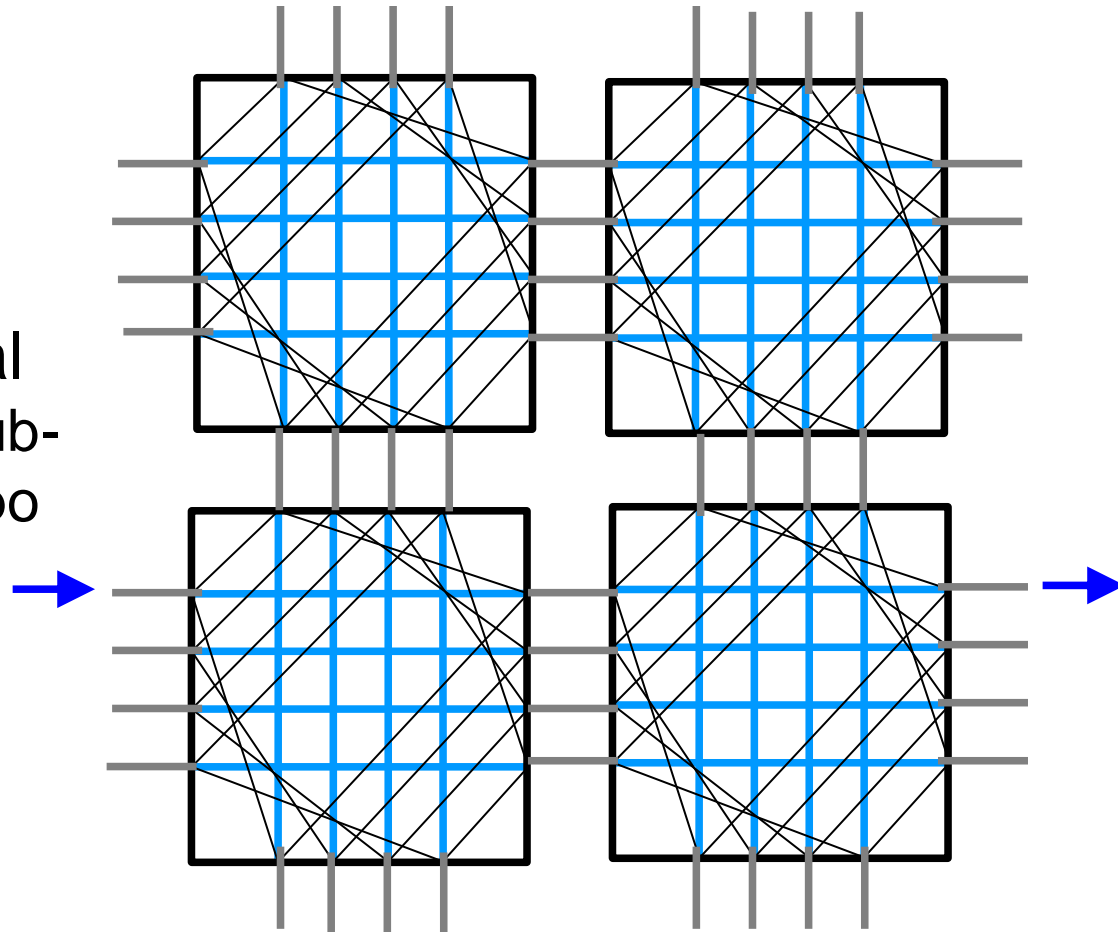
Configuração sem Conexões

Apenas um sub-elemento de conexão é testado a cada vez.



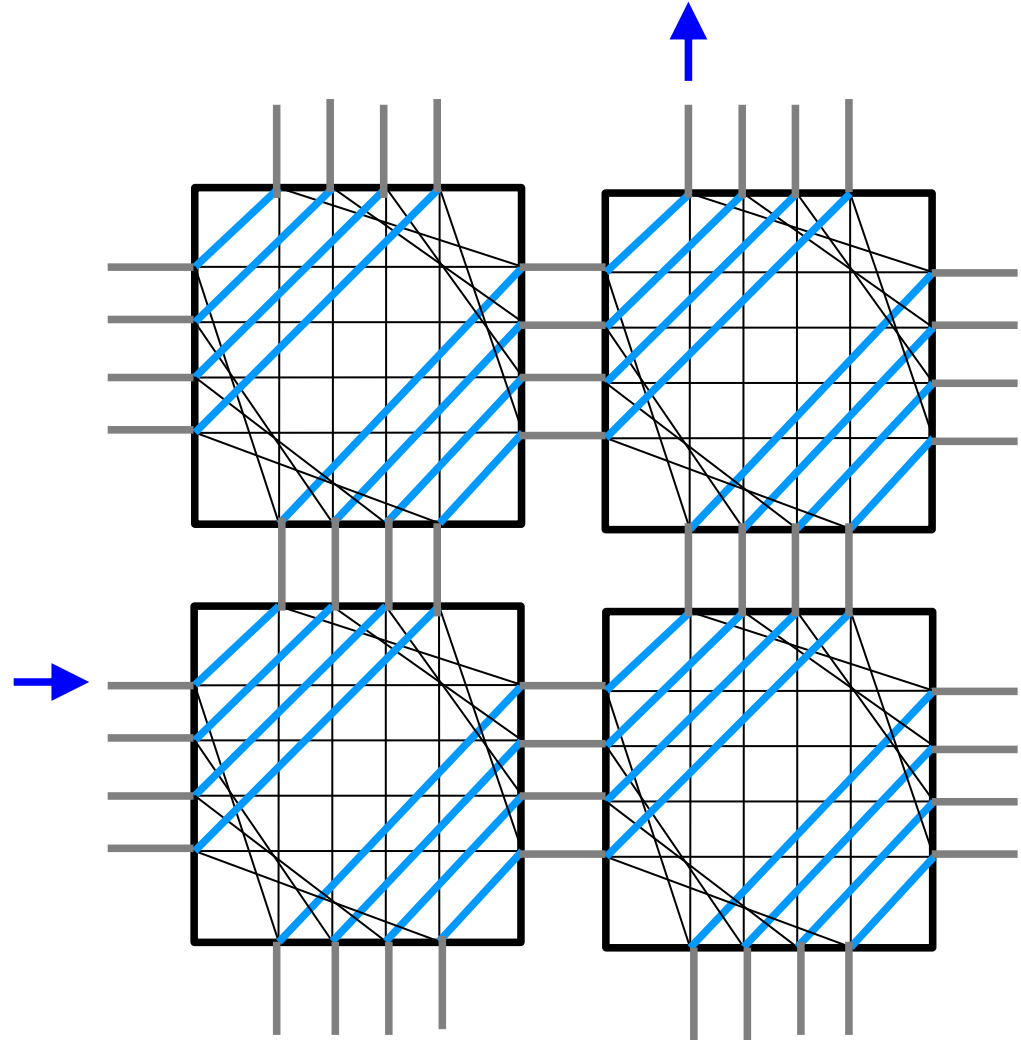
Detecção de falhas nas interconexões (2)

Configuração Ortogonal
Permite o teste de vários sub-
elementos ao mesmo tempo



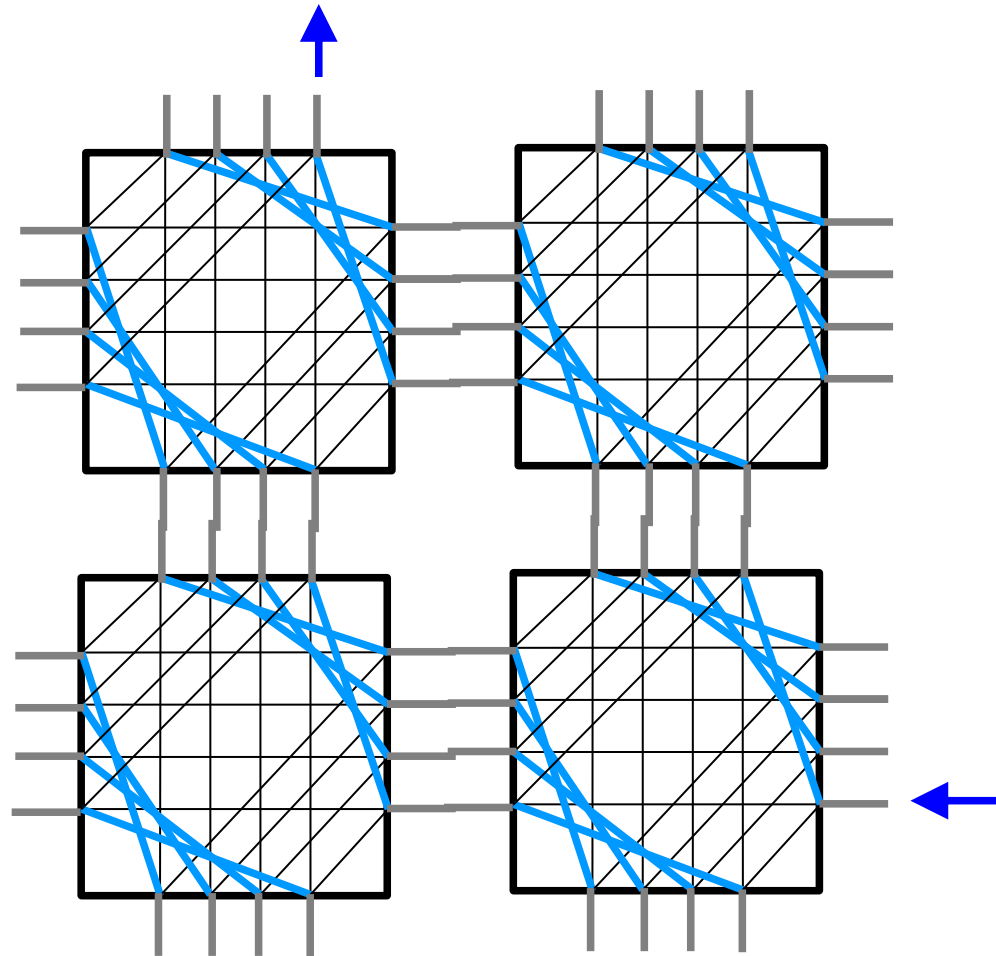
Detecção de falhas nas interconexões (3)

Configuração Diagonal 1
Permite o teste de vários sub-
elementos ao mesmo tempo



Detecção de falhas nas interconexões (4)

Configuração Diagonal 2
Permite o teste de vários sub-
elementos ao mesmo tempo



Detecção de falhas nas interconexões (5)

Vetores de Teste – 2ª Proposta:

Utilizando as configurações de teste da segunda proposta apresentada, um array completo de $m \times m$ elementos de conexão pode ser conceitualmente considerado um barramento global.

É conhecido que um barramento de n bits pode ser testado para todos curtos e abertos com $\log_2(n)$ vetores.

Então, para cada configuração apresentada, precisamos de $\log_2(2km)$ vetores de teste, resultando em um total de $3\log_2(2km)$ vetores de teste, onde k é o número de bits de cada elemento de conexão.



Teste de FPGAs

Marcelo Lubaszewski

