

# Algoritmos de Síntese Física: Testing

Gabriel Chiele  
Maiki Buffet





# Olá!

Eu sou o Maiki e este é  
o Gabriel.

Nós iremos apresentar alguns  
algoritmos de teste de síntese  
física como: *functional, scan,*  
*boundary, fault, parametric,*  
*current, wafer e memory test.*

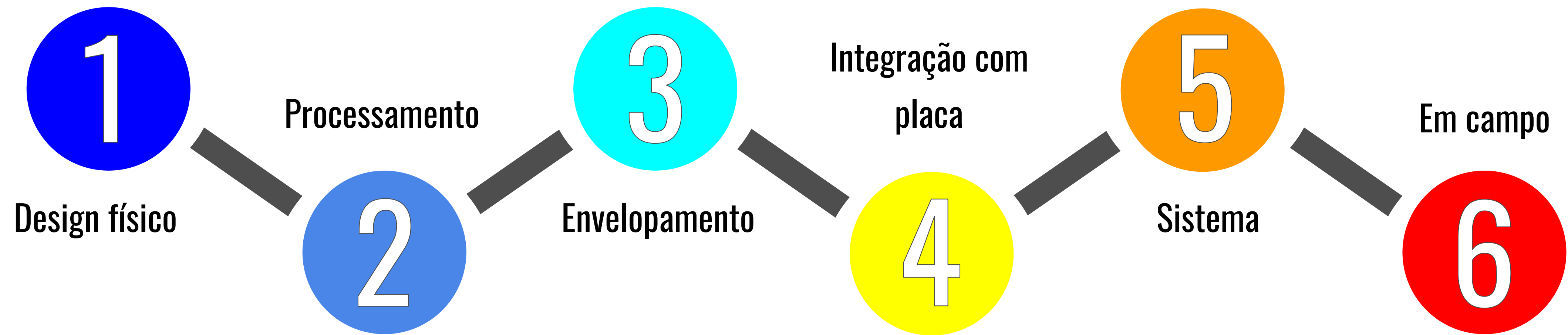
# MINIATURIZAÇÃO

O aumento na complexidade e a redução no tamanho dos dispositivos ASIC fizeram com que o teste destes dispositivos seja uma tarefa desafiadora.

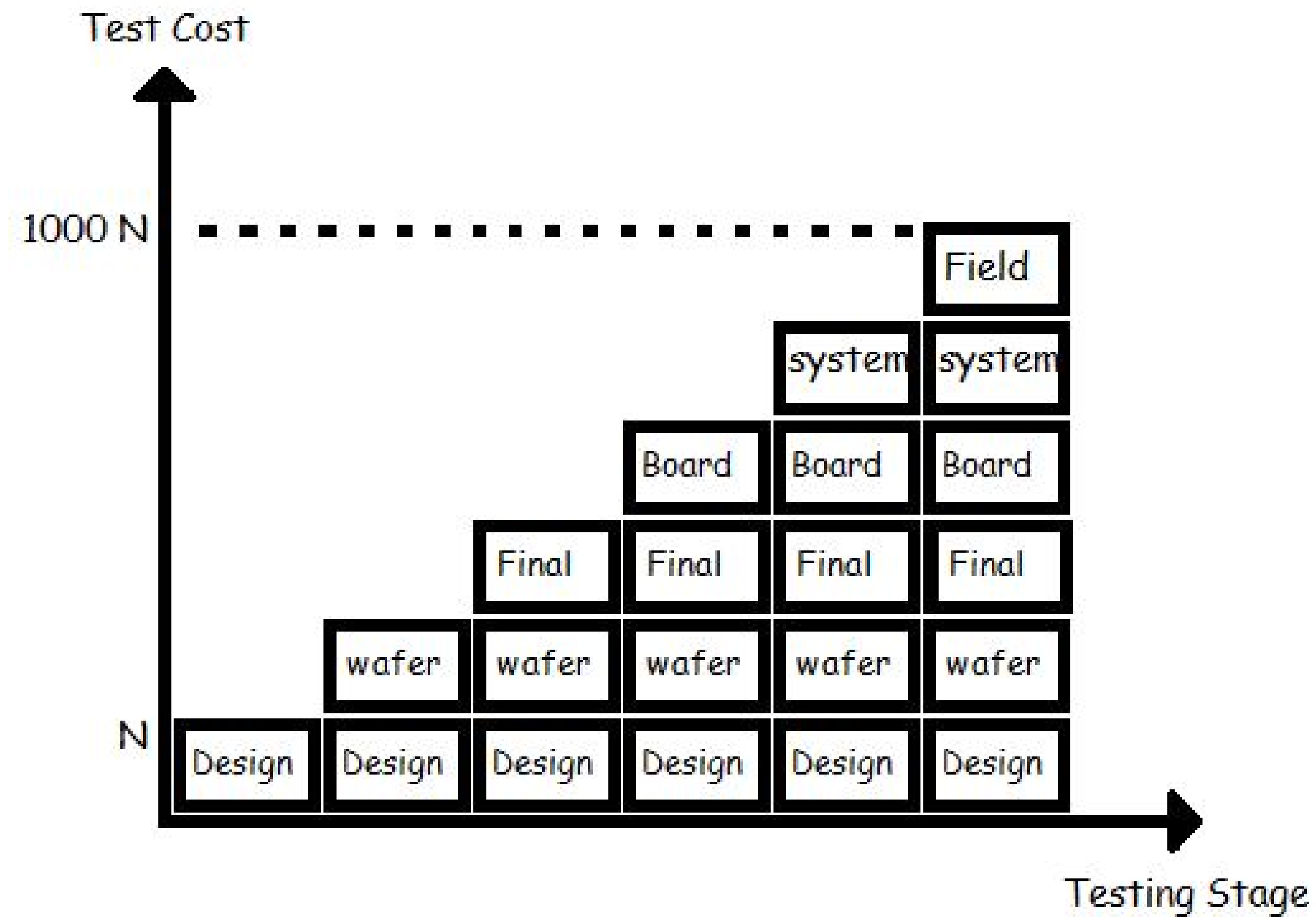
Para resolver este problema são incorporadas técnicas de *design-for-test* durante o projetos dos dispositivos.

Um requisito fundamental de produção é a habilidade de verificar o funcionamento do produto final. Sendo assim qualquer ASIC deve ser submetido a rigorosos testes durante diferentes estágio do processo.

# Processo de Teste

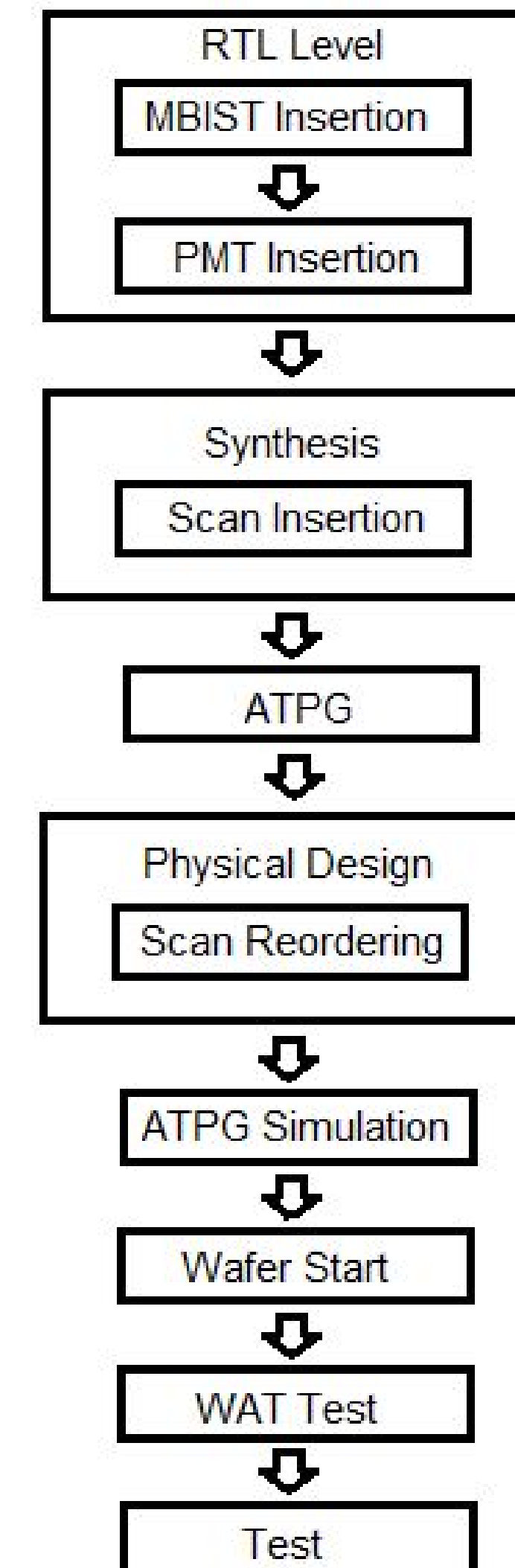


# Estágios de Teste



# Tipos de Teste

- ✓ Functional test
- ✓ Scan test
- ✓ Boundary Scan test
- ✓ Fault detection
- ✓ Parametric test
- ✓ Iddq e VLV test
- ✓ Wafer test
- ✓ Memory test
- ✓ Parallel Module test



# Functional Test

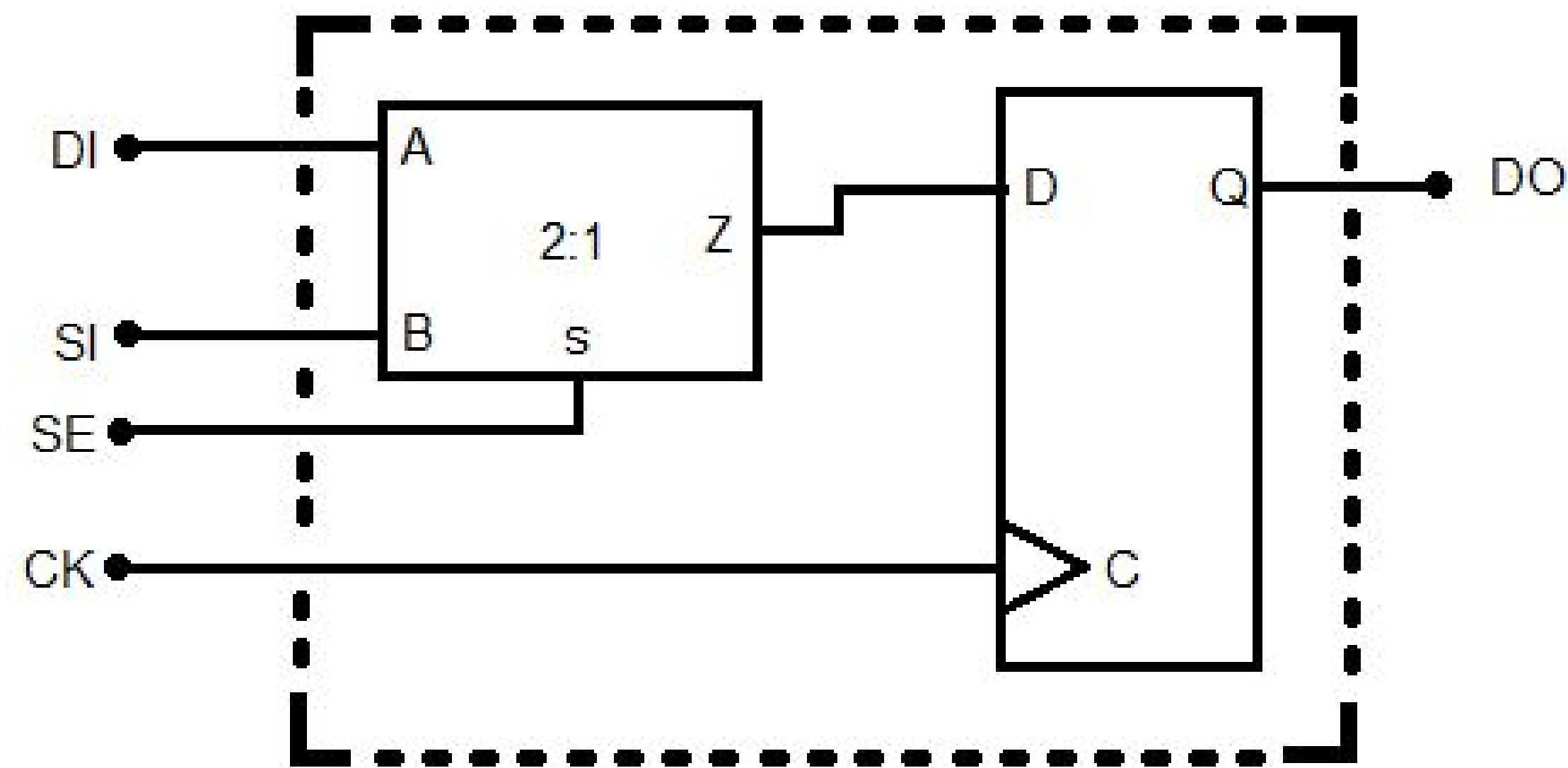
**Determina se a lógica do circuito está funcionando como deveria.**

**Necessita de um conjunto de vetores de teste para exercitar a lógica.**



# Scan Test

Detecta falhas *Stuck-At* que ocorreram durante a fabricação.



➤ Dois modos de operação:

- Normal

- Os *flip-flops* operam em paralelo com o sistema.

- Teste

- Os *flip-flops* são carregados serialmente com dados desejados.



# Boundary Scan Test

É uma extensão do método de *Scan Test* para analisar as entradas e saídas primárias do ASIC.

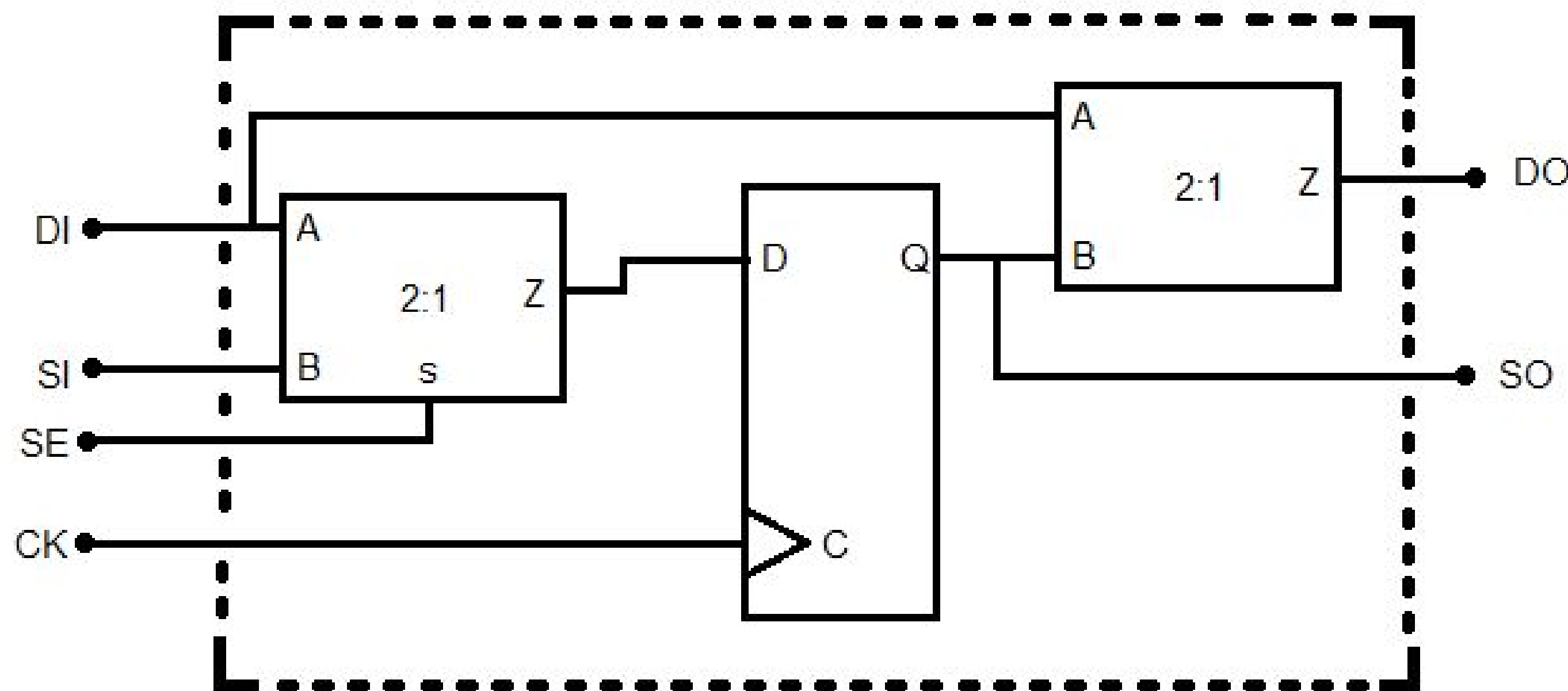
➤ Dois modos de operação:

- Normal

- Dados passam pelos pinos do ASIC entram em DI e saem em DO como se os *flip-flops* fossem transparentes.

- Teste

- Os dados passam a entrar em SI e sair em SO, assim os dados são carregados nos *flip-flops* e podem ser testados.



# Fault Detection

## Hard Defect (Stuck-At):

- Não utiliza informações reais de atraso do circuito.

- Utiliza *Automatic Test Pattern Generators*:

- Path Conditioning - Carrega valores conhecidos nos *flip-flops* de um dado caminho.
- Data capturing - Funcionamento normal.
- Data Shifting - os valores armazenados nos *flip-flops* são passados para a saída serialmente.

## Soft Defect (Slow-To):

- Utiliza informações reais de atraso e netlist do circuito.

- Utiliza Fault Simulators:

- Identifica áreas com baixa testabilidade durante o estágio de *design* comportamental.

# Parametric Test

Testa parâmetros críticos em relação a tensão AC/DC para garantir a operação na tensão de alimentação e temperatura ambiente recomendadas.

AC Test: verifica o *delay* de propagação e tensão máxima de operação.

DC Test: verifica os parâmetros DC das entradas e saídas primárias, além da corrente fornecida pela fonte de alimentação.

# Iddq e VLV Test

Detectam falhas e degradações que não podem ser detectadas por programas de teste funcional:

- Aumento na tensão de *Threshold*, aumento no delay devido a danos eletromagnéticos e defeitos em interconexões.

Não é necessário criar vetores de teste.

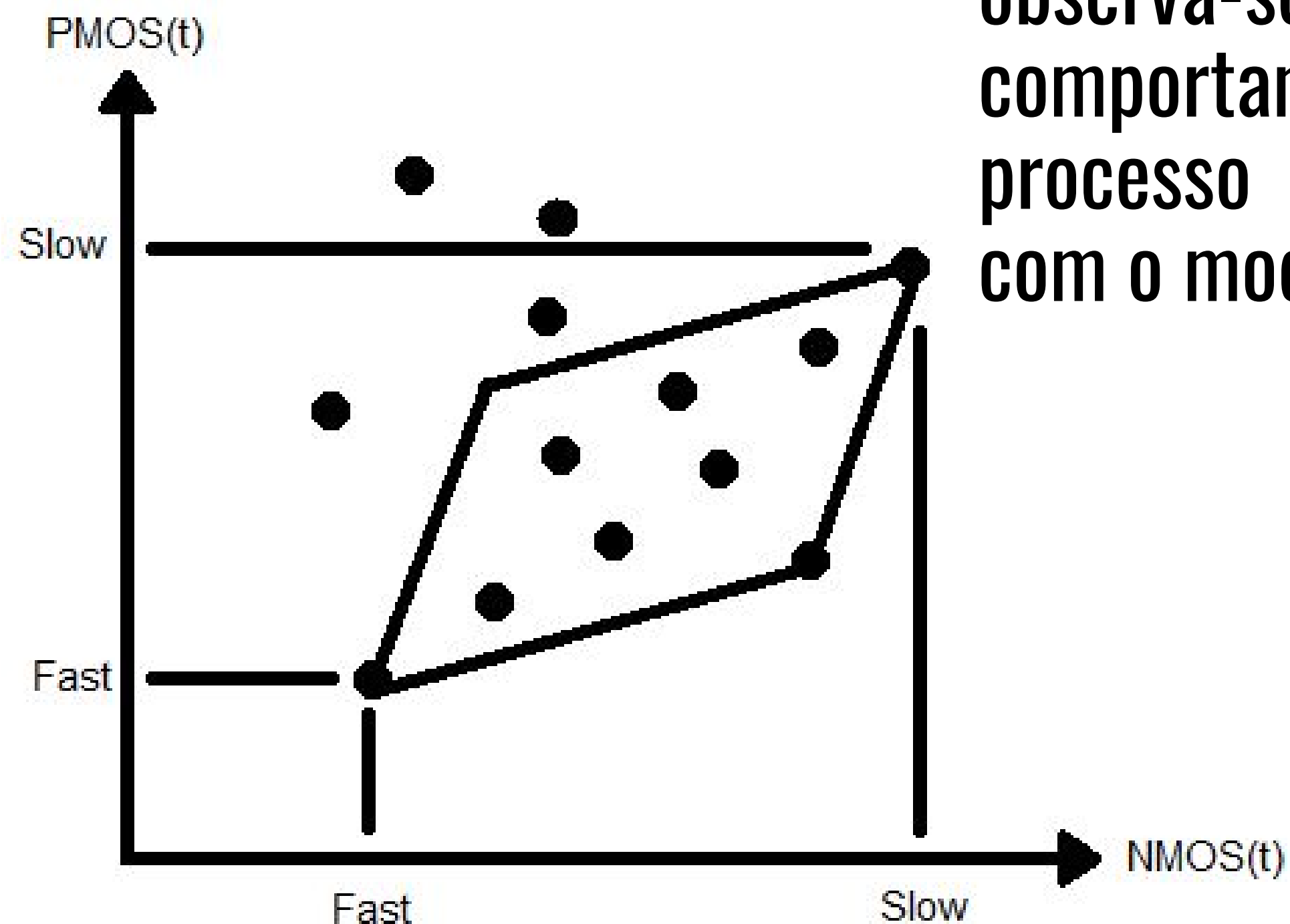
Iddq: Mede-se a corrente nas saídas do circuito e se compara a uma corrente esperada.

VLV: Aplica-se tensões baixas no circuito e se compara o resultado com um padrão esperado.



# Wafer Acceptance

Método de amostragem utilizado para determinar se o ASIC fabricado estará dentro das especificações.



Medindo os parâmetros dos PMOS e NMOS fabricados e comparando com os do modelo, observa-se se o comportamento do processo está alinhado com o modelo.

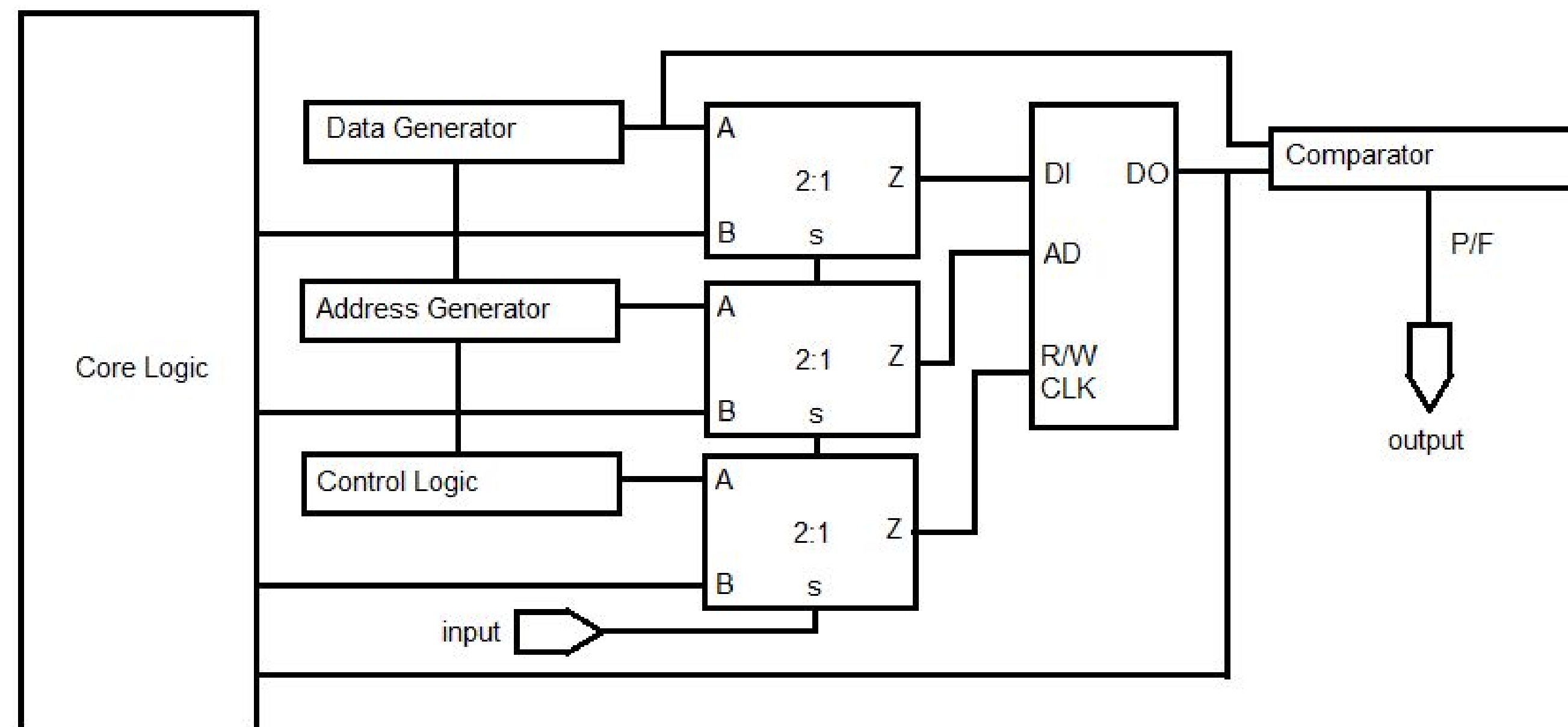
Para corrigir possíveis desalinhamentos, estende-se a região de aceitabilidade ou modifica o processo para alinhar com os modelos.

# Memory Test

Embora cause *overhead* de área no circuito, além de introduzir *delay*, é um dos métodos de teste de maior custo-benefício.

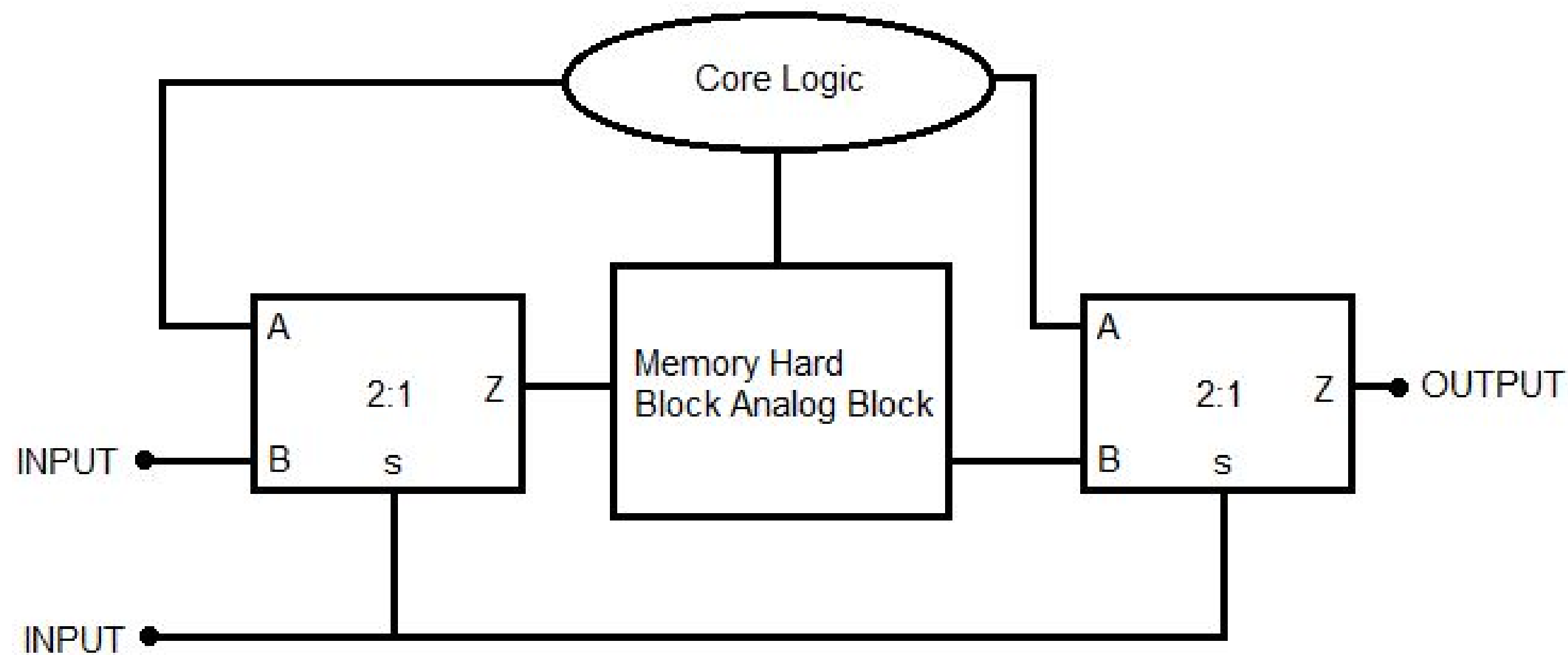
Em modo teste, os multiplexadores alteram o curso dos dados, então se exercita a memória utilizando o algoritmo *March* implementado.

*Flag* utilizada para sinalizar falha no teste.

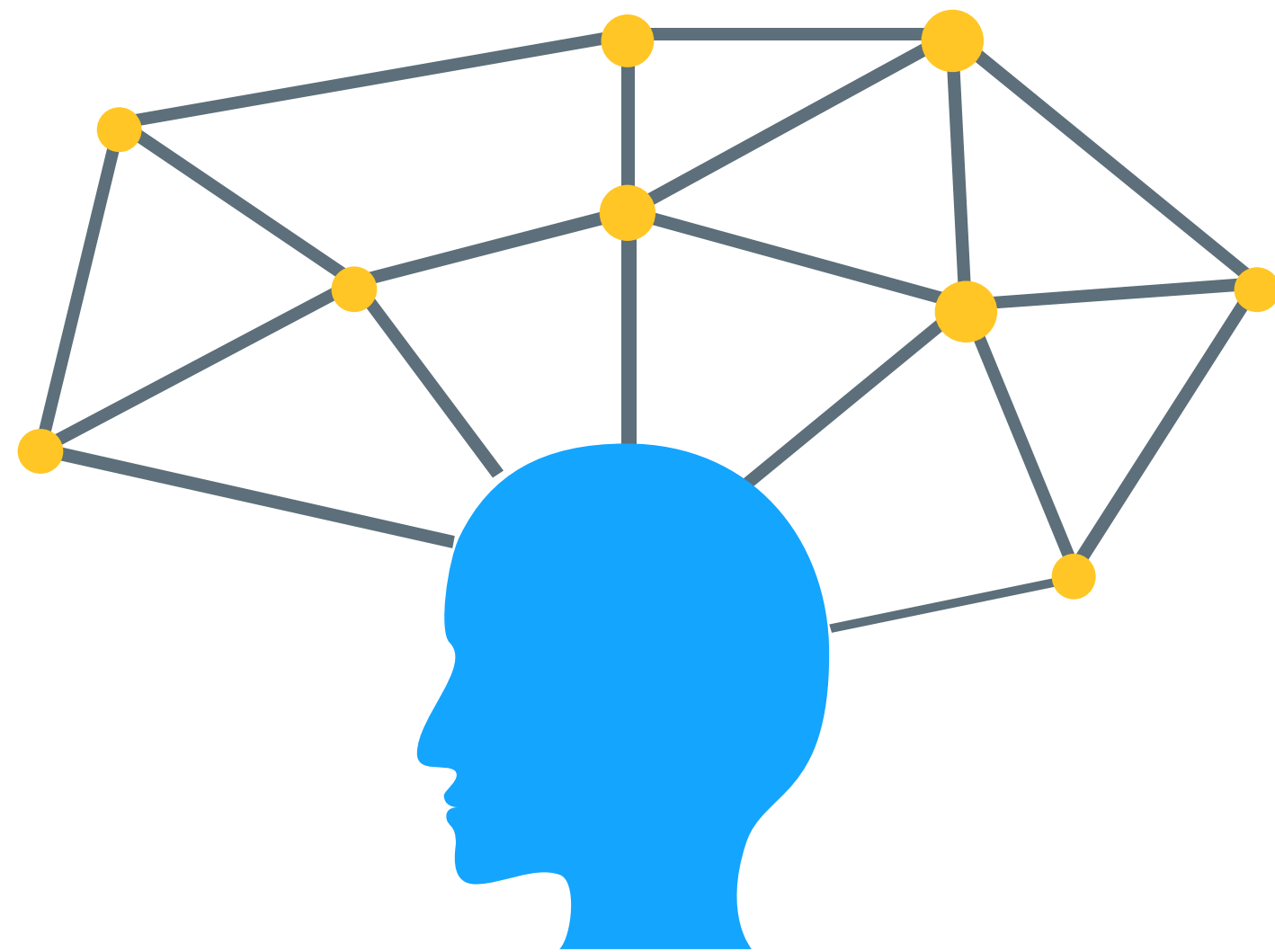


# Parallel Module Test

Aplica vetores de teste em bloco internos do *design* e observa suas respostas nas entradas e saídas primárias do ASIC.



# Implementation of March Algorithm based MBIST Architecture for SRAM



Algoritmo clássico:

(W0) >  
(R0,W1,R1,W1,R1) >  
(R1,W0,R0,W0,R0) >  
(R0,W1,R1,W1,R1) <  
(R1,W0,R0,W0,R0) <  
(R0) >

Algoritmo proposto:

(W0) >  
(R0,W1) >  
(R1) >  
(R1,W0) >  
(R0) >  
(R0,W1) <  
(R1) <  
(R1,W0) <  
(R0) <



# Resultados\*

**11x**

Na diminuição de tempo  
de teste

**1125%**

Mais rápido que o modelo  
clássico

**45**

Testes realizados durante  
quatro testes com o modelo  
clássico

\*Testes realizados em uma memória de 32 bytes



“

“ A economia  
do tempo é  
menos vulgar e  
mais  
importante que  
a do dinheiro. ”

- Marquês de Maricá







**Obrigado!**

# CRÉDITOS

- *Physical Design Essentials An ASIC Design Implementation Perspective,*  
Khosrow Golshan, Springer
- *Implementation of March Algorithm Based MBIST Architecture for SRAM,*  
M. Radha Rani, G. Rajesh Kumar, G. Prasanna Kumar, Vijetha Institute of  
Technology and Sciences, Vishnu Institute of Technology