



CI Inovador

Trilha Digital

Projeto Físico

Trabalho Final Da Disciplina

Multiplicador32FP em Ponto Flutuante Com Fluxo DFT E ATPG

Jaqueline Ferreira de Brito

Campina Grande
2025

RESUMO

Este projeto apresenta o desenvolvimento de um circuito integrado específico (ASIC) para a multiplicação de números em ponto flutuante de 32 bits, seguindo o padrão IEEE 754, com foco na implementação de técnicas de Design for Testability (DFT) e geração automática de padrões de teste (ATPG).

A arquitetura foi descrita em SystemVerilog e estruturada como uma máquina de estados finita (FSM), incorporando a detecção de condições especiais como NaN, infinito, overflow e underflow. O fluxo de desenvolvimento integrou a síntese lógica com a inserção de estruturas de teste (scan chains e controladores) utilizando a ferramenta Cadence Genus, seguida pela geração automatizada de vetores de teste estruturais com o Cadence Modus, focando em falhas do tipo stuck-at.

Os vetores de teste gerados foram exportados e validados através de simulações funcionais com a ferramenta Cadence XCelium, confirmando o correto funcionamento do circuito após a instrumentação para testabilidade. A metodologia aplicada permitiu alcançar métricas significativas de cobertura de falhas, demonstrando que a inserção das estruturas de DFT não comprometeu o desempenho do multiplicador. Os resultados obtidos evidenciam a eficácia do fluxo DFT/ATPG na criação de sistemas digitais mais robustos e testáveis.

Palavras-chave: DFT, ATPG, Multiplicador em Ponto Flutuante, IEEE 754, Testabilidade

ABSTRACT

This project presents the development of an Application-Specific Integrated Circuit (ASIC) for 32-bit floating-point multiplication, following the IEEE 754 standard, focusing on the implementation of Design for Testability (DFT) techniques and Automatic Test Pattern Generation (ATPG). The architecture was described in SystemVerilog and structured as a Finite State Machine (FSM), incorporating the detection of special conditions such as NaN, infinity, overflow, and underflow. The development flow integrated logic synthesis with the insertion of test structures (scan chains and controllers) using Cadence Genus, followed by the automated generation of structural test vectors with Cadence Modus, focusing on stuck-at fault models. The generated test vectors were exported and validated through functional simulations with Cadence XCelium, confirming the correct operation of the circuit after testability instrumentation. The applied methodology achieved significant fault coverage metrics, demonstrating that the insertion of DFT structures did not compromise the multiplier's performance. The results obtained highlight the effectiveness of the DFT/ATPG flow in creating more robust and testable digital systems.

Keywords: DFT, ATPG, Floating-Point Multiplier, IEEE 754, Testability

SUMÁRIO

1. Introdução	9
2. Especificações	10
2.1. Descrição Funcional	10
2.1.1. Visão Geral	10
2.1.2. Formato de Dados	10
2.1.3. Operação Básica	10
2.2. Blocos Funcionais e Arquitetura	10
2.2.1. Blocos Principais	10
2.2.2. Estrutura Pipeline	11
2.3. Interface do Circuito	11
2.3.1. Sinais de Entrada	12
2.3.2. Sinais de Saída	12
2.3.3. Sinais Internos Principais	12
2.4. Requisitos de Projeto	13
2.4.1. Requisitos de Desempenho	13
2.4.2. Requisitos de Alimentação	13
2.4.3. Requisitos Adicionais	13
2.5. Tecnologia de Fabricação	13
2.5.1. Processo de Fabricação	13
2.5.2. Bibliotecas e Modelos	14
2.5.3. Considerações de DFT	14
3. Metodologia	15
3.1. Visão Geral da Metodologia	15
3.1.1. Visão Geral do Fluxo de Trabalho	15
3.1.2. Estrutura de Diretórios	15
3.2. Fluxo de Trabalho e Etapas Principais	16
3.2.1. Análise Inicial do Design	16
3.2.2. Instrumentação do Design para Teste	16
3.2.3. Geração e Análise de Padrões de Teste	17
3.2.4. Geração e Verificação de Vetores de Teste	17
3.3. Ferramentas Utilizadas	18
3.3.1. Síntese Lógica com Genus	18
3.3.2. Geração de Padrões de Teste com Modus	19
4. Resultados e Discussões	21
4.1. Comparação entre os reports do Genus dos circuitos sem e com DFT	21
4.1.1 Analisando os relatórios multiplier32FP_qor.rpt e multiplier32FP_qor_dft.rpt:	21
4.1.1.1 Impacto no Timing	21
4.1.1.2 Impacto na Área	21
4.1.1.3 Impacto na Contagem de Instâncias	21
4.1.1.4 Outras Métricas	22
4.1.1.5 Conclusões Preliminares	22
4.1.2 Analisando os relatórios multiplier32FP_area.rpt e multiplier32FP_area_dft.rpt	22

4.1.2.1 Comparação das Métricas de Área: Sem DFT vs. Com DFT	22
4.1.2.2 Análise dos Resultados	23
4.1.3 Analisando os relatórios multiplier32FP_timing.rpt e multiplier32FP_timing_dft.rpt.	24
4.1.3.1 Comparação do Caminho Crítico: Sem DFT vs. Com DFT	24
4.1.3.2 Análise Detalhada do Caminho Crítico	24
4.1.3.3 Características do Caminho	24
4.1.3.4 Impacto da Inserção DFT no Timing	24
4.1.3.5 Observações Importantes	25
4.1.4 Analisando os relatórios multiplier32FP_power.rpt e multiplier32FP_power_dft.rpt.	25
4.1.4.1. Comparação do Consumo de Potência: Sem DFT vs. Com DFT	25
4.1.4.2 Distribuição por Categorias de Componentes	25
4.1.4.3 Análise dos Resultados de Potência	26
4.1.5 Analisando os relatórios multiplier32FP_gates.rpt e multiplier32FP_gates_dft.rpt	27
4.1.5.1 Comparação das Células por Tipo e Área	27
4.1.5.2 Análise Detalhada das Modificações para DFT	27
4.1.5.2.1. Transformação de Flip-flops	27
4.1.5.2.2. Aumento de Multiplexadores	28
4.1.5.2.3. Alterações Menores na Lógica Combinacional	28
4.1.5.3 Observações Importantes	28
4.2 Análise do multiplier32FP_scanDEF_dft.scandef	29
4.2.1 A estrutura implementada possui:	29
4.2.2 Distribuição Funcional dos Registradores	29
4.2.3 Análise da Estrutura de Scan	30
4.3 Análise das Diferenças entre Arquivos entre multiplier32FP.v e multiplier32FP_dft.v	31
4.3.1. Comparação entre multiplier32FP.v e multiplier32FP_dft.v	31
4.3.2. Comparação entre arquivos SDF (Standard Delay Format)	31
4.3.3 multiplier32FP_normal_worst.sdf	32
4.3.4. Impactos Práticos das Diferenças	32
4.4. Análise do log_build_model: Construção do Modelo DFT	32
4.4.1 Informações Principais do Processo	33
4.4.2 Estrutura do Modelo Construído	33
4.4.3 Otimização do Modelo para Teste	33
4.4.4 Significado para o Fluxo DFT	33
4.5 Análise do log_build_faultmodel: Modelo de Falhas do Multiplicador	34
4.5.1. Estatísticas do Modelo de Falhas	34
4.5.2. Significado dos Números	34
4.6 Análise do Modo de Teste FULLSCAN (log_build_testmode_FULLSCAN)	35
4.6.1 Configuração do Modo de Teste	35
4.6.2. Atribuição de Funções de Teste aos Pinos	36
4.6.3. Observações:	36

4.6.4. Significado para o Fluxo DFT	36
4.7 Análise da Verificação das Estruturas de Teste (log_verify_test_structures_FULLSCAN)	37
4.7.1 Resumo das Verificações Realizadas	37
4.7.2 Análise Detalhada das Cadeias de Scan	37
4.7.3. Controle e Observação	38
4.8. Análise do Processo de Construção do Modelo de Falhas	38
4.8.1. Estatísticas de Falhas Geradas	38
4.8.2. Resultados da Análise	38
4.8.3. Arquivos Gerados	39
4.9. Análise do Processo de Construção do Modelo do Circuito	39
4.9.1. Resumo da Operação	39
4.9.2. Características do Design Analisado	39
4.9.2.1. Modelo Hierárquico vs. Modelo Plano	39
4.9.2.2. Componentes do Circuito	40
4.9.3. Análise das Otimizações	40
4.9.4. Observações Técnicas	40
4.10. Análise da Construção do Modo de Teste FULLSCAN	41
4.10.1. Configuração Geral do Modo de Teste	41
4.10.1.1. Parâmetros Fundamentais	41
4.10.2. Pinos de Teste e Suas Funções	41
4.10.3. Geração Automática de Sequências	42
4.10.4. Aspectos Destacados de Testabilidade	42
4.11. Análise do Processo de Geração de Padrões de Teste ATPG	42
4.11.1. Visão Geral do Processo	42
4.11.3. Análise da Progressão da Cobertura	43
4.11.4. Resultados Finais	43
4.11.5. Conjunto Final de Padrões de Teste	43
4.11.6. Análise da Qualidade do Resultado	43
4.12. Análise das Estatísticas do Modelo do Multiplicador de Ponto Flutuante	44
4.12.1. Estrutura do Modelo	44
4.12.2. Elementos do Circuito	44
4.13. Análise Detalhada das Estruturas de Teste do Multiplicador de Ponto Flutuante	46
4.13.1. Estrutura das Cadeias de Scan	46
Visão Geral das Cadeias	46
4.13.2. Detalhamento das Cadeias de Scan	46
1. Cadeia 1 (54 bits)	46
2. Cadeia 2 (53 bits)	46
4.13.3. Análise da Organização Funcional	46
4.13.4. Implicações para Teste e Diagnóstico	47
4.14. Análise da Verificação de Estruturas de Teste (verify_test_structures)	48
4.14.1. Visão Geral do Processo de Verificação	48
4.14.2. Verificações Executadas e Resultados	48
4.14.3. Análise Detalhada das Cadeias de Scan	48

4.14.4. Conclusões da Verificação	49
4.15. Análise da Atividade de Chaveamento (write_toggle_gram)	49
4.15.1. Visão Geral da Ferramenta e Resultados	49
4.15.2. Estatísticas de Chaveamento	49
4.15.3. Sequências com Maior Chaveamento em Scan	50
4.15.4. Implicações e Preocupações	51
4.16. Análise do Processo de Geração de Vetores de Teste (write_vectors)	51
4.16.1. Visão Geral	51
4.16.2. Configuração e Parâmetros	51
4.16.3. Detalhes da Geração de Vetores	52
4.16.4. Arquivos de Saída Gerados	52
4.16.5. Conclusões da Análise	53
4.17. Análise do Processo de Geração de Vetores de Teste Temporizado (write_vectors)	53
4.17.1. Visão Geral	53
4.17.2. Parâmetros e Configuração	53
4.17.3. Detalhamento dos Vetores Gerados	54
4.17.4. Estatísticas dos Vetores de Teste	54
4.17.5. Progressão Detalhada da Cobertura	54
4.17.6. Arquivos de Saída Temporizado	55
4.17.7. Análise das Características Temporais	55
4.17.8. Conclusões da Análise	56
4.18. Análise da Simulação de Teste ATPG do Multiplicador de Ponto Flutuante	56
4.18.1. Visão Geral da Simulação	56
4.18.2. Configuração e Arquivos da Simulação	56
4.18.3. Resultados da Simulação	57
4.18.4. Análise dos Resultados	57
4.18.5. Conclusões da Análise	58
4.19. Análise Comparativa da Simulação com Temporização (FULLSCAN_TIMED)	58
4.19.1. Visão Geral	58
4.19.2. Diferenças na Configuração em Relação à Simulação Anterior	59
4.19.3. Características Exclusivas da Simulação Temporizada	59
4.19.4. Comparação de Resultados	59
4.19.5. Diferenças Estruturais na Simulação	60
a. A simulação com timing valida que os padrões funcionam não apenas logicamente, mas também respeitando as restrições de tempo	60
4.19.6. Conclusão	61
5.1. Análise dos Resultados ATPG	62
5.2. Simulação e Verificação dos Padrões	62
5.2.1. Métricas e Resultados Finais	63
5.3. Aspectos Positivos	64
5.4. Potenciais Melhorias	64
5.5. Lições Aprendidas	64
6. Referências	65

1. Introdução

A introdução apresenta o desenvolvimento de um ASIC para multiplicação de números em ponto flutuante de 32 bits (padrão IEEE 754), com foco na aplicação de técnicas de Design for Testability (DFT) e Automatic Test Pattern Generation (ATPG). O texto inicia explicando a importância da técnica de scan design, que permite alta controlabilidade e observabilidade dos elementos sequenciais através da conversão de flip-flops em registradores de deslocamento conectados em cadeia (scan chains), facilitando a aplicação de testes estruturais.

O circuito foi implementado em SystemVerilog usando uma máquina de estados finita (FSM) para gerenciar a multiplicação e detectar condições especiais (NaN, infinito, overflow e underflow). Projetado para operar a 10 MHz, o multiplicador foi verificado com um testbench self-checking e submetido a um fluxo completo de desenvolvimento que incluiu a síntese lógica e a inserção de estruturas DFT.

O objetivo principal do trabalho foi aplicar as técnicas de DFT/ATPG ao projeto, utilizando as ferramentas Cadence Genus, Modus e Xcelium para inserir scan chains, gerar vetores de teste para falhas stuck-at e validar funcionalmente os padrões gerados. O processo buscou preservar a funcionalidade original do circuito enquanto o preparava para fabricação, evidenciando como as técnicas de testabilidade aumentam a confiabilidade de sistemas digitais complexos.

2. Especificações

2.1. Descrição Funcional

2.1.1. Visão Geral

O circuito implementa um multiplicador de precisão simples para números em ponto flutuante de 32 bits, seguindo o padrão IEEE-754. O design executa multiplicação com arredondamento adequado, tratamento de exceções e casos especiais (infinitos, NaNs, zeros) conforme especificado pelo padrão.

2.1.2. Formato de Dados

Formato IEEE-754 de Precisão Simples (32 bits):

- 1 bit de sinal (bit 31)
- 8 bits de expoente com bias de 127 (bits 30-23)
- 23 bits de mantissa normalizada (bits 22-0)

2.1.3. Operação Básica

1. Extração de Componentes: Separa sinal, expoente e mantissa dos operandos
2. Cálculo do Sinal: XOR dos bits de sinal dos operandos
3. Cálculo do Expoente: Soma dos expoentes menos o bias (127)
4. Multiplicação das Mantissas: Multiplicação completa das mantissas com bit implícito
5. Normalização: Ajuste do resultado para manter a forma normalizada
6. Arredondamento: Implementação de arredondamento para o "mais próximo" (round-to-nearest)
7. Detecção de Exceções: Identificação de overflow, underflow, NaN, infinito
8. Saída Final: Composição do resultado em formato IEEE-754

2.2. Blocos Funcionais e Arquitetura

2.1.2. Blocos Principais

1. Unidade de Controle:
 - a. Coordena as operações sequenciais
 - b. Gerencia os sinais de clock e reset
 - c. Controla o fluxo de dados entre os blocos
2. Estágio de Pré-processamento:
 - a. Analisador de casos especiais
 - b. Extrator de campos (sinal, expoente, mantissa)

- c. Inserção do bit implícito na mantissa
- 3. Unidade de Multiplicação:
 - a. Multiplicador paralelo para mantissas de 24 bits (23 + bit implícito)
 - b. Circuito de pipeline para otimização de desempenho
- 4. Estágio de Pós-processamento:
 - a. Normalizador do resultado
 - b. Unidade de arredondamento
 - c. Detector de overflow/underflow
 - d. Manipulador de exceções
- 5. Registradores de Pipeline:
 - a. Registradores intermediários para implementação em pipeline
 - b. Armazenamento temporário de valores de entrada e sinais de controle

2.2.2. Estrutura Pipeline

O multiplicador é implementado com uma arquitetura pipeline de três estágios:

1. Estágio 1: Decodificação e pré-processamento
2. Estágio 2: Multiplicação das mantissas e soma dos expoentes
3. Estágio 3: Normalização, arredondamento e composição do resultado

2.3. Interface do Circuito

2.3.1. Sinais de Entrada

Sinal	Largura	Descrição
clk	1 bit	Clock do sistema
rst_n	1 bit	Reset assíncrono ativo em nível baixo
a	32 bits	Operando A em formato IEEE-754
b	32 bits	Operando B em formato IEEE-754
start	1 bit	Sinal para iniciar a multiplicação
scan_enable	1 bit	Sinal de controle do modo scan (DFT)
scan_in1	1 bit	Entrada da cadeia de scan 1 (DFT)
scan_in2	1 bit	Entrada da cadeia de scan 2 (DFT)

2.3.2. Sinais de Saída

Sinal	Largura	Descrição
result	32 bits	Resultado da multiplicação (IEEE-754)
overflow	1 bit	Flag de overflow
underflow	1 bit	Flag de underflow
inexact	1 bit	Flag de resultado inexato
invalid	1 bit	Flag de operação inválida
done	1 bit	Sinaliza conclusão da operação
scan_out1	1 bit	Saída da cadeia de scan 1 (DFT)
scan_out2	1 bit	Saída da cadeia de scan 2 (DFT)

2.3.3. Sinais Internos Principais

Sinal	Largura	Descrição
sign_a, sign_b	1 bit	Bits de sinal extraídos dos operandos
exp_a, exp_b	8 bits	Campos de expoente dos operandos
mant_a, mant_b	24 bits	Mantissas com bit implícito
mant_prod	48 bits	Produto das mantissas
exp_sum	9 bits	Soma dos expoentes (com bit extra)
result_sign	1 bit	Sinal do resultado final
result_exp	8 bits	Expoente do resultado final
result_mant	23 bits	Mantissa normalizada final

2.4. Requisitos de Projeto

2.4.1. Requisitos de Desempenho

Parâmetro	Especificação	Observações
-----------	---------------	-------------

Frequência de Operação	10 MHz	Clock de sistema
Latência	3 ciclos de clock	Tempo total para produzir resultado
Throughput	1 operação/ciclo	Após preenchimento do pipeline
Tempo de Setup	< 0.8 ns	Para operação em 500 MHz
Tempo de Hold	> 0.1 ns	Para garantir estabilidade

2.4.2. Requisitos de Alimentação

Parâmetro	Especificação	Observações
Tensão de Alimentação	1.0V \pm 10%	VDD para núcleo lógico
Potência Máxima	10 mW	Consumo sob carga máxima
Potência em Standby	< 1 mW	Quando sem operação ativa

2.4.3. Requisitos Adicionais

Requisito	Especificação
Testabilidade	Cobertura de falhas stuck-at > 99.5%
Área Máxima	0.15 mm ²
Temperatura de Operação	-40°C a 125°C
Tempo de Reset	2 ciclos de clock após ativação

2.5. Tecnologia de Fabricação

2.5.1. Processo de Fabricação

- Tecnologia: GPDK 45nm
- Biblioteca de Células: gsclib045_svt_v4.4 (Standard Voltage Threshold)
- Modelo de Transistor: Bulk CMOS
- Metais: 8 camadas de metal
- Tensão Nominal: 1.0V

2.5.2. Bibliotecas e Modelos

- Biblioteca Standard Cell: slow_vdd1v0_basicCells.v

- Modelo de Timing: Corner slow (worst-case)
- Modelos SDF: Disponíveis para verificação com timing
- Corner de Processo: Typical process, 1.0V, 25°C

2.5.3. Considerações de DFT

- Metodologia: Full Scan com cadeias de scan múltiplas
- Cobertura Mínima: 99.5% para falhas stuck-at
- Otimização: Balanceamento das cadeias de scan (diferença máxima de 1 flip-flop)
- Inserção de Scan: Substituição de todos os elementos sequenciais por células scan
- Sinais de Controle: Modo scan controlado externamente via scan_enable

Este documento de especificação define completamente o multiplicador de ponto flutuante de 32 bits, fornecendo todas as informações necessárias para sua implementação, validação e teste em conformidade com os padrões da indústria e requisitos de desempenho estabelecidos.

3. Metodologia

Metodologia DFT para o Multiplicador de Ponto Flutuante de 32 Bits

3.1. Visão Geral da Metodologia

A metodologia abrange desde o desenvolvimento do RTL até a geração e verificação dos padrões de teste, utilizando um fluxo digital completo e integrado.

Este trabalho seguiu um fluxo completo de Design-for-Testability (DFT) para um multiplicador de ponto flutuante de 32 bits. O processo foi estruturado em várias etapas sequenciais e interdependentes, garantindo uma abordagem sistemática para maximizar a testabilidade do circuito, seguindo as melhores práticas da indústria de semicondutores.

3.1.1. Visão Geral do Fluxo de Trabalho

O fluxo de trabalho é composto por cinco fases principais, organizadas sequencialmente e com verificações intermediárias:

1. Desenvolvimento e Verificação do RTL
2. Síntese Lógica com Genus
3. Inserção de Estruturas DFT com Genus
4. Geração de Padrões de Teste com Modus
5. Simulação e Verificação dos Padrões

3.1.2. Estrutura de Diretórios

A organização do projeto segue uma estrutura hierárquica claramente definida:

```
multiplier32FP/  
├── frontend/                                # Projeto RTL original e verificação  
│   ├── multiplier32FP.sv                  # Código RTL em SystemVerilog  
│   ├── multiplier32FP_tb.sv              # Testbench para verificação  
│   └── run_sim.sh                         # Script para execução da simulação  
└── backend/                               # Implementação física  
    ├── synthesis/                         # Síntese lógica  
    ├── scripts/                          # Scripts de automação  
    │   ├── common/                      # Scripts comuns para configuração  
    │   │   ├── path.tcl                 # Definição de caminhos  
    │   │   └── tech.tcl                 # Configuração de tecnologia  
    │   ├── multiplier32FP.tcl           # Script de síntese  
    │   ├── multiplier32FP_dft.tcl       # Script de inserção DFT  
    │   └── modus.sh                     # Script para ATPG  
    ├── constraints/                      # Restrições de temporização  
    └── multiplier32FP.sdc                # Restrições SDC
```

```
|— reports/           # Relatórios gerados pelos tools
|— deliverables/      # Netlists e arquivos SDF finais
|— work/              # Área de trabalho
    |— test_scripts/  # Scripts e resultados de teste
    |   |— testresults/ # Resultados dos testes ATPG
    |   |   |— logs/    # Logs de execução
    |   |   |— verilog/ # Vetores de teste
    |   |— tbdata/     # Dados de teste intermediários
    |— fv/             # Verificação formal
```

3.2. Fluxo de Trabalho e Etapas Principais

3.2.1. Análise Inicial do Design

1. Linguagem: SystemVerilog
2. Arquivo Principal: frontend/multiplier32FP.sv
3. Avaliação do RTL Original:
 - a. Identificação de elementos sequenciais
 - b. Análise de controlabilidade e observabilidade
 - c. Identificação de pontos críticos para testabilidade
4. Planejamento da Estratégia DFT:
 - a. Seleção da técnica de scan (full-scan)
 - b. Definição do número e configuração das cadeias de scan
 - c. Determinação dos sinais de controle necessários

3.2.2. Instrumentação do Design para Teste

1. Inserção de Estruturas de Scan:
 - a. Substituição de flip-flops normais por células scan
 - b. Criação de duas cadeias de scan balanceadas (54 e 53 bits)
 - c. Conexão dos sinais de scan_in, scan_out e scan_enable
2. Verificação da Estrutura de Scan:
 - a. Uso do comando report_test_structures para validação
 - b. Verificação da integridade das cadeias e conectividade
 - c. Confirmação da ausência de avisos e erros estruturais

3.2.3. Geração e Análise de Padrões de Teste

1. Configuração do Ambiente ATPG:
 - a. Definição do modelo de falhas (stuck-at)
 - b. Configuração dos parâmetros de geração de padrões
 - c. Estabelecimento de metas de cobertura (>99.5%)
2. Geração de Padrões:
 - a. Uso das ferramentas Cadence Modus para ATPG

- b. Geração de padrões de teste para cadeias de scan
- c. Geração de padrões de teste para lógica combinacional
- 3. Análise de Cobertura:
 - a. Avaliação da cobertura atingida (99.95%)
 - b. Análise das falhas não testáveis
 - c. Otimização dos padrões para máxima eficiência
- 4. Análise de Potência:
 - a. Avaliação do chaveamento durante operações de scan e captura
 - b. Identificação de pontos de alta atividade
 - c. Alertas para possíveis problemas de dissipação térmica

3.2.4. Geração e Verificação de Vetores de Teste

- 1. Conversão para Formatos de Simulação:
 - a. Geração de vetores Verilog para simulação
 - b. Criação de formatos regulares e temporizados
 - c. Configuração de arquivos de suporte (cyclemap, script principal)
- 2. Simulação Funcional:
 - a. Simulação Verilog dos padrões sem considerações de timing
 - b. Validação da aplicação lógica dos padrões
 - c. Verificação da detecção correta de falhas
- 3. Simulação com Timing:
 - a. Simulação com anotação SDF para inclusão de atrasos reais
 - b. Validação dos padrões sob restrições de timing
 - c. Confirmação de operação correta com temporização real

3.3. Ferramentas Utilizadas

Ferramenta	Versão	Propósito
Cadence Genus	N/D	Síntese lógica e inserção DFT
Cadence Modus	22.12-s028_1	Geração de padrões ATPG
Cadence Xcelium	24.09-s001	Simulação e verificação
GPDK045	4.4	Biblioteca de tecnologia (45nm)

3.3.1. Síntese Lógica com Genus

- 1. Preparação do Ambiente
 - a. Ferramenta: Cadence Genus Synthesis Solution
 - b. Tecnologia: GPDK045 (45nm)
 - c. Biblioteca de Células: Standard Voltage Threshold (SVT)

2. Os arquivos de configuração incluem:
 - a. Tech Setup: backend/synthesis/scripts/common/tech.tcl
 - b. Path Setup: backend/synthesis/scripts/common/path.tcl
 - c. Constraints: backend/synthesis/constraints/multiplier32FP.sdc
3. 4.2 Execução da Síntese
 - a. Script Principal: backend/synthesis/scripts/multiplier32FP.tcl
4. Parâmetros de Síntese:
 - a. Clock: 10 MHz (período de 100ns)
 - b. Esforço: Médio
 - c. Otimização: Área, timing e potência
5. Análise dos Resultados da Síntese com e sem DFT
 - a. Relatórios Gerados:
 - i. Área: reports/multiplier32FP_area.rpt
 - ii. Timing: reports/multiplier32FP_timing.rpt
 - iii. Potência: reports/multiplier32FP_power.rpt
 - iv. QoR: reports/multiplier32FP_qor.rpt
 - b. Arquivos de Saída:
 - i. Netlist: deliverables/multiplier32FP.v
 - ii. SDF: deliverables/multiplier32FP_delays.sdf
 - iii. SDC: reports/multiplier32FP_sdc.sdc
6. Inserção de Estruturas DFT com Genus
 - a. Planejamento DFT
 - i. Estratégia: Scan completo (full-scan)
 - ii. Estilo de Scan: Muxed-scan
 - b. Configuração:
 - i. Duas cadeias de scan balanceadas
 - ii. Comprimento máximo de 54 flip-flops por cadeia
 - iii. Total de 107 elementos sequenciais
7. Execução da Inserção DFT
 - a. Script Principal: backend/synthesis/scripts/multiplier32FP_dft.tcl
 - b. Sinais de Controle:
 - c. scan_in: Entrada de scan
 - d. scan_out: Saída de scan
 - e. SE: Habilitação de scan (scan enable)
8. Análise dos Resultados da Inserção DFT
 - a. Relatórios Específicos de DFT:
 - i. Área: reports/multiplier32FP_area_dft.rpt
 - ii. Timing: reports/multiplier32FP_timing_dft.rpt
 - iii. Potência: reports/multiplier32FP_power_dft.rpt
 - iv. QoR: reports/multiplier32FP_qor_dft.rpt
 - v. Scan Definition: reports/multiplier32FP_scanDEF_dft.scandef
9. Arquivos de Saída:
 - a. Netlist com DFT: deliverables/multiplier32FP_dft.v

- b. SDF com DFT: deliverables/multiplier32FP_dft_delays.sdf
- c. SDC com DFT: reports/multiplier32FP_sdc_dft.sdc

3.3.2. Geração de Padrões de Teste com Modus

1. Preparação para ATPG
 - a. Ferramenta: Cadence Modus DFT Software Solution (v22.12-s028_1)
 - b. Script de Automação: backend/synthesis/scripts/modus.sh
 - c. Arquivo de Netlist: work/test_scripts/multiplier32FP.test_netlist.v
 - d. Tecnologia:


```
/pdk/cadence/GPDK045/gsclib045_svt_v4.4/gsclib045/verilog/slow_vdd1v0_basicCells.v
```
2. Sequência de Comandos ATPG
 - a. Construção do Modelo (build_model)
 - i. Compilação do netlist com DFT
 - ii. Carregamento da biblioteca de tecnologia
 - b. Configuração do Modo de Teste (build_testmode)
 - i. Modo: FULLSCAN
 - ii. Atribuição de pinos: multiplier32FP.FULLSCAN.pinassign
 - c. Verificação de Estruturas (verify_test_structures)
 - i. Validação das cadeias de scan
 - ii. Análise de conectividade DFT
 - d. Construção do Modelo de Falhas (build_faultmodel)
 - i. Tipo: Stuck-at (preso em 0/1)
 - ii. Falhas dinâmicas: Desabilitadas
 - e. Geração de Padrões (create_logic_tests)
 - i. Modo: FULLSCAN
 - ii. Experimento: multiplier32FP_atpg
 - f. Análise de Chaveamento (write_toggle_gram)
 - i. Avaliação da atividade de comutação
 - ii. Identificação de potenciais hotspots de potência
 - g. Exportação de Vetores (write_vectors)
 - i. Formatos: Verilog, STIL, WGL, TDL
 - ii. Variantes: Normal e com timing (TIMED)

Esta metodologia representa um fluxo completo, eficiente e robusto para a implementação de DFT em um multiplicador de ponto flutuante, garantindo alta testabilidade e confiabilidade para fabricação.

4. Resultados e Discussões

4.1. Comparação entre os reports do Genus dos circuitos sem e com DFT

4.1.1 Analisando os relatórios multiplier32FP_qor.rpt e multiplier32FP_qor_dft.rpt:

4.1.1.1 Impacto no Timing

Métrica	Sem DFT	Com DFT	Diferença	Impacto (%)
Período do Clock	100000.0 ps	100000.0 ps	0	0%
Critical Path Slack	97127.8 ps	97114.8 ps	-13.0 ps	-0.013%
TNS	0.0	0.0	0	0%
Violações de Timing	0	0	0	0%

O impacto no timing é mínimo, com uma redução de apenas 13 ps no slack do caminho crítico, o que representa aproximadamente 0,013%. Ambos os designs mantêm uma folga de aproximadamente 97 ns em um período de clock de 100 ns, indicando que o circuito opera bem abaixo da frequência máxima possível.

4.1.1.2 Impacto na Área

Métrica	Sem DFT	Com DFT	Diferença	Aumento (%)
Área de Células	4473.360	4780.134	306.774	6.86%
Área de Interconexão	2415.856	2532.799	116.943	4.84%
Área Total	6889.216	7312.933	423.717	6.15%

A inserção de estruturas DFT resultou em um aumento de 6,15% na área total do design. Este overhead é considerado moderado e está dentro do esperado para circuitos com DFT implementado.

4.1.1.3 Impacto na Contagem de Instâncias

Tipo de Instância	Sem DFT	Com DFT	Diferença	Aumento (%)
Instâncias Leaf Total	1556	1608	52	3.34%
Instâncias Sequenciais	107	107	0	0%
Instâncias Combinacionais	1449	1501	52	3.59%

Observações importantes:

- O número de instâncias sequenciais permaneceu o mesmo (107), o que é esperado, pois a técnica DFT modifica os flip-flops existentes em vez de adicionar novos
- O aumento ocorreu apenas nas instâncias combinacionais (+52), correspondendo aos multiplexadores de scan inseridos nas cadeias
- Estes 52 novos componentes representam aproximadamente um multiplexador por flip-flop na cadeia de scan, confirmando a implementação do modo scan

4.1.1.4 Outras Métricas

Métrica	Sem DFT	Com DFT	Diferença
Fanout Médio	2.5	2.6	+0.1
Relação Terms:Net	3.5213	3.5851	+0.0638
Relação Terms:Instance	4.3066	4.3632	+0.0566

O ligeiro aumento nas métricas de fanout e relações de terms indica a complexidade adicional de roteamento introduzida pelos componentes DFT.

4.1.1.5. Conclusões Preliminares

A implementação DFT no multiplicador de ponto flutuante de 32 bits resultou em:

1. Impacto mínimo no timing: Uma redução de apenas 0,013% no slack do caminho crítico
2. Overhead moderado de área: Um aumento de 6,15% na área total do design
3. Aumento esperado de instâncias: 52 instâncias combinacionais adicionais (3,59%)

Estes resultados mostram que a implementação DFT foi eficiente, mantendo o desempenho temporal praticamente inalterado enquanto adiciona testabilidade ao design com um overhead de área aceitável.

4.1.2 Analisando os relatórios multiplier32FP_area.rpt e multiplier32FP_area_dft.rpt

4.1.2.1 Comparação das Métricas de Área: Sem DFT vs. Com DFT

Métrica		Sem DFT	Com DFT	Diferença Absoluta	Aumento (%)
Contagem de Células		1556	1608	+52	3.34%
Área de Células		4473.360	4780.134	+306.774	6.86%
Área de Interconexão		2415.856	2532.799	+116.943	4.84%
Área Total		6889.216	7312.933	+423.717	6.15%

4.1.2.2 Análise dos Resultados

1. Overhead de Área:

- A implementação DFT resultou em um aumento de 6.15% na área total do design
- Este overhead é mais significativo na área das células (6.86%) do que na área de interconexão (4.84%)

2. Relação com a Contagem de Células:

- Foram adicionadas 52 células ao design, representando um aumento de 3.34%
- O aumento percentual na área (6.15%) é maior que na contagem de células (3.34%), o que sugere que as células adicionadas para DFT têm área média maior que as células do design original

3. Interpretação do Overhead:

- O overhead de área observado está dentro do intervalo típico esperado para implementações DFT (5-10%)
- A proporção entre área de células e área de interconexão se manteve aproximadamente a mesma, indicando que a inserção DFT não alterou significativamente as características de roteamento do design

O overhead de área de 6.15% representa um custo relativamente baixo considerando os benefícios de testabilidade obtidos com a implementação DFT.

4.1.3 Analisando os relatórios multiplier32FP_timing.rpt e multiplier32FP_timing_dft.rpt.

4.1.3.1 Comparação do Caminho Crítico: Sem DFT vs. Com DFT

Métrica	Sem DFT	Com DFT	Diferença	Impacto (%)
Período de Clock	100000 ps	100000 ps	0	0%
Critical Path Slack	97128 ps	97115 ps	-13 ps	-0.013%
Setup Time	16 ps	29 ps	+13 ps	+81.25%
Data Path Delay	2532 ps	2532 ps	0	0%

4.1.3.2 Análise Detalhada do Caminho Crítico

Em ambos os casos, o caminho crítico é idêntico:

- Ponto inicial: Entrada a_i[2]
- Ponto final: Registrador mant_prod_r_reg[47]/D
- Natureza do caminho: Multiplicação de mantissa (parte mais complexa de um multiplicador de ponto flutuante)

4.1.3.3 Características do Caminho

1. Estrutura similar: O caminho atravessa principalmente células de adição (ADDFX1) que formam a árvore de multiplicação
2. Mesmo atraso de dados: O data path delay permanece inalterado (2532 ps) entre as duas implementações
3. Diferença principal: O flip-flop destino foi alterado de um DFFRX2 (sem DFT) para um SDFFRHGX2 (com DFT)

4.1.3.4 Impacto da Inserção DFT no Timing

1. Aumento do setup time: O tempo de setup aumentou em 13 ps (81.25%), devido à complexidade adicional do flip-flop com capacidade de scan
2. Impacto mínimo no slack total: Redução de apenas 0.013% no slack do caminho crítico
3. Sem alteração no padrão de propagação: O caminho de dados permanece intacto, mostrando que a inserção DFT não alterou a estrutura lógica do design

4.1.3.5 Observações Importantes

1. Margem de timing ampla: Ambos os designs têm slacks extremamente grandes (aproximadamente 97 ns em um período de clock de 100 ns), indicando que o circuito opera muito abaixo de sua frequência máxima potencial.
2. Eficiência da implementação DFT: O impacto negligenciável no timing demonstra que a estratégia DFT foi implementada de forma eficiente, adicionando testabilidade com mínimo impacto no desempenho.
3. Células escaneáveis: A substituição dos flip-flops normais por células escaneáveis (SDFF) é visível no relatório, mas seu impacto no timing é mínimo devido ao grande slack disponível.

A análise de timing confirma que a implementação DFT não compromete significativamente o desempenho temporal do multiplicador de ponto flutuante, mantendo praticamente o mesmo caminho crítico e alterando o slack total em menos de 0.02%.

4.1.4 Analisando os relatórios multiplier32FP_power.rpt e multiplier32FP_power_dft.rpt.

4.1.4.1. Comparação do Consumo de Potência: Sem DFT vs. Com DFT

Categoria de Potência	Sem DFT	Com DFT	Diferença	Aumento (%)
Potência Total	6.62042e-05 W	6.64117e-05 W	+2.075e-07 W	+0.31%
Leakage	6.03765e-07 W	6.37702e-07 W	+3.394e-08 W	+5.62%
Internal	2.61175e-05 W	2.57163e-05 W	-4.012e-07 W	-1.54%
Switching	3.94829e-05 W	4.00576e-05 W	+5.747e-07 W	+1.46%

4.1.4.2 Distribuição por Categorias de Componentes

Componentes	Sem DFT	Com DFT	Diferença	Variação (%)
--------------------	----------------	----------------	------------------	---------------------

Register	1.34227e-05 W (20.27%)	1.31580e-05 W (19.81%)	-2.647e-07 W -1.97%
Logic	5.22221e-05 W (78.88%)	5.26944e-05 W (79.35%)	+4.723e-07 W +0.90%
Clock	5.59310e-07 W (0.84%)	5.59310e-07 W (0.84%)	0 W 0%

4.1.4.3 Análise dos Resultados de Potência

1. Impacto Mínimo Total:

- A inserção de DFT resultou em um aumento desprezível de apenas 0,31% na potência total, demonstrando que a implementação foi extremamente eficiente em termos de consumo energético.

2. Componentes do Aumento de Potência:

- O maior aumento percentual ocorreu na potência de leakage (+5,62%), o que é esperado devido à adição dos multiplexadores de scan que aumentam a área de silício.
- A potência de switching aumentou ligeiramente (+1,46%) devido às chaves adicionais nos caminhos de scan.
- Curiosamente, houve uma pequena redução na potência interna (-1,54%), possivelmente devido a otimizações nas células sequenciais durante a inserção DFT.

3. Distribuição por Categorias:

- A potência de registradores diminuiu ligeiramente (-1,97%), possivelmente devido às diferentes características das células de scan versus flip-flops regulares.
- A potência lógica aumentou levemente (+0,90%) devido aos multiplexadores adicionais.
- A distribuição percentual entre componentes permaneceu praticamente inalterada, mantendo a mesma proporção.

4. Eficiência da Implementação:

- O overhead de potência de 0,31% está bem abaixo do típico para implementações DFT (que frequentemente ficam entre 2-5%).
- A manutenção do perfil de distribuição de potência indica que a estrutura básica do design foi preservada.

O resultado da análise de potência é notavelmente positivo, mostrando que a testabilidade foi adicionada com um custo energético mínimo, o que é particularmente importante para aplicações sensíveis ao consumo de energia.

4.1.5 Analisando os relatórios multiplier32FP_gates.rpt e multiplier32FP_gates_dft.rpt

4.1.5.1 Comparação das Células por Tipo e Área

Categoria	Sem DFT	Com DFT	Diferença	Variação (%)
Total de Instâncias	1556	1608	+52	+3.34%
Área Total	4473.360	4780.134	+306.774	+6.86%
Células Sequenciais	107 (763.002)	107 (953.838)	0 (+190.836)	0% (+25.01%)
Células de Lógica	1347 (3640.248)	1395 (3752.766)	+48 (+112.518)	+3.56% (+3.09%)
Inversores	102 (70.110)	106 (73.530)	+4 (+3.420)	+3.92% (+4.88%)

4.1.5.2 Análise Detalhada das Modificações para DFT

4.1.5.2.1. Transformação de Flip-flops

Tipo de Flip-flop	Sem DFT	Com DFT
DFFRHQX1	51 (313.956)	0
DFFRX1/X2	5 (35.568)	0
DFFSHQX1	3 (19.494)	0
SDDFRHQX1	0	64 (525.312)
SDDFRHQX2	0	40 (396.720)
SDDFSXL	0	3 (31.806)

Todos os flip-flops normais (DFF) foram substituídos por suas versões escaneáveis (SDFF), que incorporam a funcionalidade de scan. Apesar de manter o

mesmo número de flip-flops (107), a área ocupada aumentou 25%, refletindo a complexidade adicional das células escaneáveis.

4.1.5.2.2. Aumento de Multiplexadores

Tipo de Multiplexador	Sem DFT	Com DFT	Diferença
MX2X1	78 (186.732)	125 (299.250)	+47 (+112.518)
Outros MUXs	30 (96.444)	30 (96.444)	0

O aumento significativo de multiplexadores 2:1 (MX2X1) é a evidência mais clara da implementação DFT, onde estes multiplexadores são inseridos para selecionar entre dados funcionais normais e dados da cadeia de scan.

4.1.5.2.3. Alterações Menores na Lógica Combinacional

Algumas pequenas alterações na lógica combinacional foram necessárias para acomodar os requisitos de DFT:

- Aumento de 4 inversores
- Adição de 1 porta NOR2BX1
- Adição de 1 porta NOR4BX1
- Adição de 1 inversor INVX3 (mais potente)

4.1.5.3 Observações Importantes

1. Distribuição da Área:

- Sequencial: aumentou de 17.1% para 20.0% da área total
- Lógica: reduziu de 81.4% para 78.5% da área total
- Esta mudança na proporção reflete o maior overhead imposto pelos elementos sequenciais com DFT

2. Eficiência da Implementação:

- A inserção DFT foi realizada de forma eficiente, mantendo a integridade estrutural do design
- O acréscimo de 52 instâncias (3.34%) é extremamente próximo do esperado para este tipo de design, onde tipicamente são inseridos multiplexadores de scan para cada flip-flop
- A implementação feita com primitivas SDFF (scan D flip-flops) integradas ajudou a minimizar o overhead

3. Overhead Controlado:

- O overhead de área total de 6.86% está dentro do limite aceitável para implementações DFT

- b. O aumento é principalmente atribuível às células de flip-flop escaneáveis maiores e aos multiplexadores adicionais.

4.2 Análise do multiplier32FP_scanDEF_dft.scandef

O arquivo multiplier32FP_scanDEF_dft.scandef define a implementação física das cadeias de scan no multiplicador de ponto flutuante, revelando como os registradores foram organizados para permitir testabilidade completa.

4.2.1 A estrutura implementada possui:

Característica	Cadeia 1	Cadeia 2	Total
Nome	top_chain_seg1_clk_rising	AutoChain_2_seg1_clk_rising	-
Tamanho máximo	54 bits	53 bits	107 bits
Pino entrada	scan_in	dft_sdi_1	-
Pino saída	scan_out	dft_sdo_1	-
Partição	p_clk_rising_best	p_clk_rising_best	-

4.2.2 Distribuição Funcional dos Registradores

1. Cadeia 1 (54 registradores):
 - a. Contadores e flags de controle:
 - i. done_delay_counter_reg[0:1] - Contadores de atraso após conclusão
 - ii. done_r_reg - Flag de conclusão
 - iii. initial_counter_reg[0:3] - Contadores de inicialização
 - b. Expoente:
 - i. exp_sum_r_reg[0:9] - Registradores do resultado da operação com expoentes
 - c. Flags especiais:
 - i. infinit_r_reg - Flag para resultados infinitos

- d. Mantissa (primeira parte):
 - i. `mant_prod_r_reg[0:35]` - Primeira parte dos registradores da mantissa do produto (36 bits)
- 2. Cadeia 2 (53 registradores):
 - a. Mantissa (parte final):
 - i. `mant_prod_r_reg[36:47]` - Restante dos registradores da mantissa (12 bits)
 - b. Resultado final:
 - i. `product_r_reg[0:31]` - Registradores do produto final em formato IEEE-754
 - c. Flags de status:
 - i. `nan_r_reg` - Flag para Not-a-Number
 - ii. `overflow_r_reg` - Flag de overflow
 - iii. `underflow_r_reg` - Flag de underflow
 - iv. `zero_sujo_r_reg` - Flag de zero "sujo" (denormalizado)
 - d. Controle de arredondamento:
 - i. `round_toward_zero_r_reg` - Modo de arredondamento para zero
 - e. Máquina de estados:
 - i. `state_r_reg[0:3]` - Registradores do estado atual do multiplicador

4.2.3 Análise da Estrutura de Scan

1. Balanceamento Eficiente: As duas cadeias foram balanceadas de forma quase perfeita (54/53 bits), minimizando o tempo de aplicação de vetores e otimizando a área.
2. Agrupamento Lógico: Os registradores foram agrupados de maneira lógica em cada cadeia:
 - a. Cadeia 1: Foco em registradores de controle e primeira parte da mantissa
 - b. Cadeia 2: Foco no produto final, flags de status e estado da máquina
3. Cobertura Completa: Todos os 107 flip-flops do design foram incluídos nas cadeias de scan, garantindo testabilidade completa para falhas stuck-at e de transição.
4. Interface DFT Dedicada: A implementação usa quatro pinos dedicados para teste (`scan_in`, `scan_out`, `dft_sdi_1`, `dft_sdo_1`), além dos pinos de controle SE (Shift Enable) e `rst_n` (como modo de teste).

4.3 Análise das Diferenças entre Arquivos entre multiplier32FP.v e multiplier32FP_dft.v

4.3.1. Comparação entre multiplier32FP.v e multiplier32FP_dft.v

Aspecto	multiplier32FP.v	multiplier32FP_dft.v
Portas (interface)	Portas funcionais básicas (a_i, b_i, product_o, clk, rst_n)	Portas adicionais para DFT (scan_in, scan_out, dft_sdi_1, dft_sdo_1, SE)
Flip-flops	Células padrão (DFFRHQX1, DFFRX1, etc.)	Células escaneáveis (SDFFRHQX1, SDFFRHQX2, etc.)
Multiplexadores	Apenas multiplexadores funcionais	Multiplexadores adicionais para cadeias de scan
Estrutura interna	Estrutura direta da implementação funcional	Estrutura modificada com caminhos alternados para modo de teste
Declarações	Apenas declarações funcionais	Declarações adicionais para controle de scan

4.3.2. Comparação entre arquivos SDF (Standard Delay Format)

multiplier32FP_delays.sdf vs multiplier32FP_dft_delays.sdf

Aspecto	multiplier32FP_delays.sdf	multiplier32FP_dft_delays.sdf
Células contempladas	Apenas células funcionais	Células funcionais + células de scan
Caminhos de atraso	Caminhos funcionais	Caminhos funcionais + caminhos de scan
Valores de setup/hold	Valores para flip-flops normais	Valores para flip-flops escaneáveis (geralmente maiores)
Atrasos de rede	Atrasos para conexões funcionais	Atrasos para conexões funcionais e de scan

Tamanho do Menor
arquivo

Maior (devido aos caminhos
adicionais)

4.3.3 multiplier32FP_normal_worst.sdf

Contém os piores casos de atraso para o design original (sem DFT) em condições extremas de processo, tensão e temperatura (PVT). É usado para verificações de timing no pior caso para garantir que o circuito funcione corretamente mesmo nas condições mais adversas.

4.3.4. Impactos Práticos das Diferenças

1. Simulação:
 - a. O netlist com DFT requer simulações em dois modos: funcional e teste.
 - b. Os arquivos SDF com DFT possuem informações de timing mais complexas.
2. Verificação Funcional:
 - a. O design com DFT precisa ser verificado tanto no modo funcional quanto no modo de teste.
 - b. É necessário validar que a inserção DFT não afeta a funcionalidade normal.
3. Análise de Timing:
 - a. O circuit com DFT geralmente tem requisitos de timing mais restritivos
 - b. Os caminhos críticos podem ser diferentes entre as versões.
4. Layout Físico:
 - a. A versão com DFT requer mais área de silício e possivelmente estratégias de roteamento diferentes.
 - b. Os pinos de teste precisam ser acessíveis para conexão com equipamento ATE.

Estas diferenças são essenciais para entender como a testabilidade foi implementada no design e como isso afeta o desempenho, área e comportamento do circuito.

4.4. Análise do log_build_model: Construção do Modelo DFT

O arquivo de log “*log_build_model*” documenta o processo de construção do modelo do circuito para análise de testabilidade. Este é o passo inicial e fundamental para todo o fluxo DFT, pois define a estrutura do design que será utilizada para simulação e geração dos vetores de teste.

4.4.1 Informações Principais do Processo

- Ferramenta: Cadence Modus DFT Software Solution, versão 22.12-s028_1
- Netlist analisado: multiplier32FP.test_netlist.v (versão com estruturas de teste)
- Biblioteca de tecnologia: GPDK045 (GlobalFoundries 45nm)

4.4.2 Estrutura do Modelo Construído

1. Estatísticas do Modelo Hierárquico
 - a. Blocos: 9.191
 - b. Pinos: 28.889
 - c. Redes (nets): 16.276
2. Estatísticas do Modelo Plano (Flattened)
 - a. Blocos/nós: 7.677
 - b. Entradas primárias: 70
 - c. Saídas primárias: 41
3. Elementos Sequenciais
 - a. Flip-flops total: 107
 - i. 104 com porta de reset (Rising Edge Flop w/Reset)
 - ii. 3 com porta de set (Rising Edge Flop w/Set)
 - b. Sem latches no design
4. Outras Estatísticas Importantes
 - a. Instâncias de células da biblioteca: 1.608
 - b. Redes amarradas (tied nets): 107 total
 - i. 104 amarradas a '0'
 - ii. 3 amarradas a '1'

4.4.3 Otimização do Modelo para Teste

Durante o processo de construção do modelo para teste, algumas otimizações foram realizadas:

- Remoção de lógica para 3 de 58 células
- Eliminação de 856 nós lógicos pendentes (dangling logic nodes)

Estas otimizações são importantes para simplificar o modelo sem afetar a testabilidade, removendo estruturas redundantes ou inacessíveis que não contribuem para a cobertura de falhas.

4.4.4 Significado para o Fluxo DFT

Este modelo serve como base para todas as etapas subsequentes do fluxo DFT:

1. Será usado para analisar os possíveis pontos de falha (fault sites)
2. Permitirá a verificação das estruturas de teste (scan chains)
3. Será a base para a geração dos padrões de teste via ATPG

A qualidade deste modelo determina a precisão e eficiência dos vetores de teste gerados posteriormente.

4.5 Análise do log_build_faultmodel: Modelo de Falhas do Multiplicador

O arquivo log_build_faultmodel documenta a criação do modelo de falhas para o multiplicador de ponto flutuante, que é um passo crucial para determinar quantas falhas precisam ser testadas e para servir de base para a geração de vetores de teste.

4.5.1. Estatísticas do Modelo de Falhas

1. Falhas Estáticas (Stuck-At)
 - a. Total de falhas estáticas: 14.254
 - b. Falhas estáticas colapsadas: 9.368
 - c. Taxa de colapsamento: 34,3% (redução de 4.886 falhas)
 - d. Falhas em entradas primárias (PI): 140
 - e. Falhas em saídas primárias (PO): 82
2. Outras Categorias de Falhas
 - a. Falhas dinâmicas: 0 (não incluídas conforme especificado por -includedynamic no)
 - b. Falhas IDDq: 14.254 (falhas detectáveis por medição de corrente)

4.5.2. Significado dos Números

1. Alto Número de Falhas:
 - a. 14.254 falhas totais indicam a complexidade do design do multiplicador
 - b. Cada nó interno e conexão pode sofrer falhas stuck-at-0 e stuck-at-1
2. Eficiência do Colapsamento de Falhas:
 - a. O colapsamento reduziu o número total de falhas em 34,3%
 - b. Isso significa que 9.368 falhas são "únicas" do ponto de vista comportamental
 - c. Esta é uma otimização crítica para tornar o processo ATPG mais eficiente
3. Distribuição de Falhas:
 - a. Com 140 falhas em PIs e 82 em POs, a grande maioria das falhas (14.032) está em nós internos
 - b. Isso ressalta a importância das cadeias de scan para acessar nós internos que, de outra forma, seriam difíceis de testar
4. Cobertura Inicial Zero:

- a. Neste estágio, a cobertura é 0% porque ainda não foram gerados padrões de teste
- b. O objetivo das etapas posteriores será maximizar esta cobertura

4.5.3 Impacto no Fluxo DFT

Este modelo de falhas serve como referência contra a qual todos os vetores de teste serão avaliados. Algumas implicações importantes:

1. Definição do Objetivo de Cobertura:
 - a. O alvo é cobrir o máximo possível das 9.368 falhas colapsadas
 - b. Típicamente, uma cobertura acima de 95% é considerada excelente para designs complexos
2. Complexidade do ATPG:
 - a. O número de falhas influencia diretamente o tempo e os recursos necessários para geração de padrões
 - b. O colapsamento de falhas foi uma otimização essencial, reduzindo em aproximadamente 1/3 o esforço necessário
3. Base para Avaliação da Qualidade dos Vetores:
 - a. Qualquer vetor de teste gerado será avaliado por sua eficácia em detectar essas falhas
 - b. A cobertura final determinará a qualidade da implementação DFT

Este modelo de falhas estabelece a base para as próximas etapas do fluxo DFT, principalmente a geração de vetores de teste ATPG e a avaliação da cobertura de falhas.

4.6 Análise do Modo de Teste FULLSCAN (log_build_testmode_FULLSCAN)

4.6.1 Configuração do Modo de Teste

O log documenta a criação do modo de teste FULLSCAN, que define como os vetores de teste serão aplicados e como os resultados serão observados:

Característica	Configuração
Tipo de Scan	GSD (Generalized Scan Design)
Boundary Scan	NONE (Sem boundary scan)
Entradas	PI (Primary Inputs)
Saídas	PO (Primary Outputs)
Cobertura lógica	100.00% ativa, 0% inativa, 0% restrita

Número de cadeias 2 (controlável e observável)

4.6.2. Atribuição de Funções de Teste aos Pinos

A configuração definida estabelece como os diferentes pinos do circuito serão usados durante o teste:

Função	Contagem	Pinos Designados
Clocks do Sistema (SC)	2	rst_n (+), clk (-)
Clocks de Deslocamento (EC)	1	clk (-)
Habilitação de Scan (SE)	1	SE
Entradas de Scan (SI)	2	scan_in, dft_sdi_1
Saídas de Scan (SO)	2	scan_out, dft_sdo_1

4.6.3. Observações:

- O sinal clk serve tanto como clock do sistema quanto clock de deslocamento, com polaridade negativa
- O sinal rst_n funciona como um clock do sistema com polaridade positiva
- O pino SE controla o modo de operação (funcional vs. teste)

4.6.4. Significado para o Fluxo DFT

Esta configuração de modo de teste é fundamental para:

1. Controlar a aplicação de vetores de teste:
 - a. Define como os padrões são carregados nas cadeias de scan
 - b. Estabelece a sequência de inicialização e operação durante o teste
2. Maximizar a testabilidade:
 - a. 100% da lógica está ativa no modo de teste
 - b. As duas cadeias de scan estão completamente controláveis e observáveis
3. Sincronização do teste:
 - a. O uso coordenado de clocks do sistema e de deslocamento permite sequenciar corretamente:
 - i. Carregamento de vetores de teste
 - ii. Captura de respostas
 - iii. Deslocamento de dados nas cadeias de scan

Este modo de teste bem configurado garante que os vetores ATPG possam ser aplicados corretamente ao circuito, permitindo alta cobertura de falhas e teste eficiente.

4.7 Análise da Verificação das Estruturas de Teste (log_verify_test_structures_FULLSCAN)

Este log documenta os resultados da ferramenta verify_test_structures, que valida a integridade e a funcionalidade das estruturas DFT implementadas no multiplicador de ponto flutuante.

4.7.1 Resumo das Verificações Realizadas

A ferramenta executa diversas verificações críticas para garantir que a implementação DFT está correta:

Verificação	Resultado	Observações
Controle de clocks nos elementos de memória	Aprovado	Sem problemas relatados
Contenção em drivers tri-state	Aprovado	Sem problemas relatados
Loops de feedback e dispositivos keeper	Aprovado	Sem problemas relatados
Choppers de clock	Aprovado	Nenhum identificado
Características dos flip-flops e latches	Aprovado	Todas as cadeias validadas
Latches de valor fixo explícito	Aprovado	Nenhum identificado
Race conditions potenciais em sinais de clock	Aprovado	Sem problemas relatados
Integridade das cadeias de scan internas	Aprovado	Todas as cadeias verificadas

4.7.2 Análise Detalhada das Cadeias de Scan

1. Estatísticas das Cadeias
 - a. Número total de cadeias: 2
 - b. Total de bits em cadeias: 107
 - c. Comprimento médio das cadeias: 53,5 bits

- d. Comprimento da cadeia mais longa: 54 bits
 - e. Balanceamento das cadeias: 101% (muito próximo ao ideal)
 - f. Características de Cada Cadeia
2. Cadeia 1:
 - a. Entrada: pino scan_in
 - b. Saída: pino scan_out
 - c. Comprimento: 54 bits
 - d. Status: Controlável e observável
 3. Cadeia 2:
 - a. Entrada: pino dft_sdi_1
 - b. Saída: pino dft_sdo_1
 - c. Comprimento: 53 bits
 - d. Status: Controlável e observável

4.7.3. Controle e Observação

- Cadeias controláveis por entradas primárias (SI): 2
- Cadeias observáveis por saídas primárias (SO): 2
- Cadeias controláveis por geradores de padrões on-chip: 0
- Cadeias observáveis por registradores MISR on-chip: 0

4.8. Análise do Processo de Construção do Modelo de Falhas

O log apresentado documenta a execução bem-sucedida do comando `build_faultmodel`, que é responsável por analisar o design do multiplicador de ponto flutuante e criar um modelo completo de falhas para teste.

4.8.1. Estatísticas de Falhas Geradas

Tipo de Falha	Total	Colapsadas	Status
Falhas Estáticas	14.254	9.368	100% ativas
Falhas em PIs	140	-	100% ativas
Falhas em POs	82	-	100% ativas
Falhas IDDq	14.254	-	100% ativas
Falhas Dinâmicas	0	0	(não incluídas)

4.8.2. Resultados da Análise

1. Eficiência do Colapsamento:
 - a. O processo conseguiu reduzir o número de falhas a serem testadas em 34,3%

- b. De 14.254 falhas totais para 9.368 falhas colapsadas
- 2. Acessibilidade Completa:
 - a. 100% das falhas estão ativas, o que indica que todas podem potencialmente ser testadas
 - b. Não existem falhas inativas que seriam impossíveis de testar
- 3. Distribuição das Falhas:
 - a. Apenas 1,6% das falhas estão nas entradas/saídas primárias ($140 + 82 = 222$)
 - b. 98,4% das falhas estão em nós internos, ressaltando a importância das cadeias de scan
- 4. Qualidade da Execução:
 - a. A execução foi rápida (0.02s de tempo decorrido)
 - b. Não foram reportados avisos ou erros durante o processo

4.8.3. Arquivos Gerados

O processo criou dois arquivos importantes:

- faultModel: Contém a descrição completa de todas as falhas possíveis no circuito
- faultStatus: Registra o status atual de cada falha (testada, não testada, etc.)

Este modelo de falhas servirá como base para o próximo passo do fluxo DFT, que é a geração de padrões de teste ATPG. A alta qualidade do modelo de falhas e a ausência de problemas durante sua construção são indicadores positivos para uma boa cobertura de teste nas etapas subsequentes.

4.9. Análise do Processo de Construção do Modelo do Circuito

4.9.1. Resumo da Operação

O log documenta a execução do comando `build_model`, que representa a etapa inicial e fundamental do fluxo DFT, onde a ferramenta Cadence Modus analisa o netlist do circuito e constrói um modelo interno para todas as operações subsequentes de teste.

4.9.2. Características do Design Analisado

4.9.2.1. Modelo Hierárquico vs. Modelo Plano

Característica	Modelo Hierárquico	Modelo Plano (Flattened)
Blocos	9.191	7.677

Pinos/Nós	28.889 pinos	7.677 nós
Redes (Nets)	16.276	-

4.9.2.2. Componentes do Circuito

Tipo de Componente	Quantidade	Detalhamento
Entradas Primárias	70	Todas input-only
Saídas Primárias	41	Todas output-only
Flip-flops	107	104 com reset, 3 com set
Células da Biblioteca	1.608	GPDK045 (45nm)
Redes Amarradas	107	104 a '0', 3 a '1'

4.9.3. Análise das Otimizações

Durante o processo de construção do modelo, o sistema realizou automaticamente algumas otimizações:

- Remoção de lógica para 3 de 58 células no design
- Eliminação de 856 nós lógicos pendentes (dangling logic nodes)
- Preservação total dos elementos sequenciais (107 flip-flops)

4.9.4. Observações Técnicas

1. Consistência com Análises Anteriores:
 - a. O número de flip-flops (107) corresponde exatamente às duas cadeias de scan (54+53)
 - b. O total de 1.608 células corresponde ao inventário de células lógicas que vimos no relatório de área
2. Considerações sobre Avisos:
 - a. Os três avisos reportados se referem a pinos sem conexão em células da biblioteca (LATCH_udp_2, altos_dff_r, altos_dff_s)
 - b. Estes são avisos benignos relacionados à definição da biblioteca, não à implementação do design
3. Eficiência do Processo:
 - a. Tempo CPU: apenas 0.24 segundos
 - b. Memória utilizada: aproximadamente 104 MB

Este modelo construído com sucesso serve como fundação para todo o fluxo DFT subsequente, incluindo construção do modelo de falhas, análise de testabilidade e geração dos padrões ATPG.

4.10. Análise da Construção do Modo de Teste FULLSCAN

4.10.1. Configuração Geral do Modo de Teste

O log apresenta a definição e configuração do modo de teste FULLSCAN para o multiplicador de ponto flutuante, realizada através do comando `build_testmode`. Este processo é essencial para estabelecer como os padrões de teste serão aplicados ao circuito.

4.10.1.1. Parâmetros Fundamentais

Parâmetro	Configuração	Significado
Tipo de Scan	GSD	Generalized Scan Design - metodologia padrão para DFT
Boundary Scan	NONE	Sem uso de boundary scan (JTAG)
Entrada/Saída	PI/PO	Uso de entradas/saídas primárias para controle de teste

4.10.2. Pinos de Teste e Suas Funções

A ferramenta identificou e configurou os seguintes pinos para funções específicas de teste:

Função	Qtd	Pinos Designados	Polaridade
Clocks do Sistema (SC)	2	rst_n, clk	+, -
Clock Externo (EC)	1	clk	-
Habilitação de Scan (SE)	1	SE	+
Entradas de Scan (SI)	2	scan_in, dft_sdi_1	N/A
Saídas de Scan (SO)	2	scan_out, dft_sdo_1	N/A

- Observações importantes:
 - O sinal clk desempenha dupla função: clock do sistema e clock externo, ambos com polaridade negativa

- O sinal rst_n foi configurado como clock do sistema com polaridade positiva
- Não há pinos de isolamento de clock ou inibição de teste, simplificando o controle

4.10.3. Geração Automática de Sequências

A ferramenta gerou automaticamente duas sequências críticas:

1. Sequência de inicialização do modo (modeinit): Controla a transição do circuito do modo funcional para o modo de teste
2. Sequência de operação de scan (scanop): Define as operações de deslocamento e captura durante o teste

4.10.4. Aspectos Destacados de Testabilidade

- Atividade Lógica: 100.00% de lógica ativa, sem lógica inativa ou restrita
- Cadeias de Scan: As 2 cadeias implementadas estão totalmente controláveis e observáveis
- Latches Especiais: Não foram necessárias células de latch especiais para teste

Este modo de teste FULLSCAN bem configurado permite o controle e observabilidade necessários para a aplicação eficaz dos vetores de teste, sendo uma base sólida para alcançar alta cobertura de falhas.

4.11. Análise do Processo de Geração de Padrões de Teste ATPG

4.11.1. Visão Geral do Processo

O log documenta a execução do comando create_logic_tests, que realiza a geração automática de padrões de teste (ATPG) para o multiplicador de ponto flutuante. Este é um passo crítico que determina a qualidade final da implementação DFT.

4.11.2. Progresso da Geração de Padrões e Cobertura

Fases da Geração de Testes

Fase de Geração	Padrões	Falhas Detectadas	Cobertura Acumulada
Scan	1	868	6.09%
Reset/Set	2	108 (105+3)	6.85%

Lógica iterrações)	Estática	(15 249	13.271	99.97%
Total		252	14.247	99.97%

4.11.3. Análise da Progressão da Cobertura

O processo mostrou um padrão de detecção de falhas muito eficiente:

- Primeiras 16 iterrações: 75.40% de cobertura (9.772 falhas)
- Iterrações 16-64: 93.46% de cobertura (adicional de 2.574 falhas)
- Iterrações 64-231: 99.85% de cobertura (adicional de 1.784 falhas)
- Iterrações finais: 99.97% de cobertura (últimas 117 falhas)

4.11.4. Resultados Finais

Estatísticas de Falhas

- Total de falhas: 14.254
- Falhas testadas: 14.247
- Falhas redundantes: 5
- Falhas não testadas: 2
- Cobertura de teste (TCov): 99.95%
- Cobertura de teste ajustada (ATCov): 99.99%

4.11.5. Conjunto Final de Padrões de Teste

- Total de sequências de teste: 252
 - Testes de scan: 1
 - Testes lógicos: 251

4.11.6. Análise da Qualidade do Resultado

1. Cobertura Excepcional:
 - a. A cobertura de 99.95% é extremamente alta para um design complexo
 - b. Apenas 2 falhas permaneceram não testadas entre 14.254 falhas possíveis
 - c. A cobertura ajustada de 99.99% (excluindo falhas redundantes) é praticamente perfeita
2. Eficiência dos Padrões:
 - a. Proporção de detecção média: ~56,5 falhas por padrão
 - b. Os primeiros padrões são altamente eficientes (detectando centenas de falhas cada)

- c. Padrões posteriores focam em falhas difíceis de detectar (hard-to-detect faults)
- 3. Performance Computacional:
 - a. Tempo de CPU: apenas 0.75 segundos
 - b. Tempo decorrido: 1.83 segundos
 - c. Memória utilizada: ~10MB

Esta geração de padrões ATPG produziu um conjunto de testes extremamente eficaz, alcançando cobertura quase perfeita com um número razoável de padrões. A implementação DFT no multiplier32FP pode ser considerada excelente com base nesses resultados.

4.12. Análise das Estatísticas do Modelo do Multiplicador de Ponto Flutuante

4.12.1. Estrutura do Modelo

O relatório `report_model_statistics` apresenta uma visão detalhada da estrutura do circuito multiplicador de ponto flutuante, mostrando suas características fundamentais que impactam tanto a testabilidade quanto o comportamento geral do design.

4.12.2 Comparação entre Modelos Hierárquico e Plano

Característica	Modelo Hierárquico	Modelo Plano
Blocos	9.191	7.677
Pinos/Nós	28.889	7.677
Redes (Nets)	16.276	-

Esta redução de aproximadamente 16% no número de blocos durante o achatamento (flattening) reflete a otimização realizada para análise de testabilidade.

4.12.2. Elementos do Circuito

- 1. Interface do Circuito
 - a. Entradas primárias: 70 (todas unidirecionais)
 - b. Saídas primárias: 41 (todas unidirecionais)
 - c. Total de I/Os: 111
- 2. Componentes Sequenciais
 - a. Flip-flops com reset: 104
 - b. Flip-flops com set: 3
 - c. Total de flip-flops: 107
 - d. Latches: 0 (ausência benéfica para testabilidade)

3. Células e Redes

- a. Instâncias de células da biblioteca: 1.608
- b. Redes amarradas a valores fixos: 107
 - i. 104 amarradas a '0'
 - ii. 3 amarradas a '1'

4. Componentes Especiais (Ausentes)

- a. Sem choppers de clock
- b. Sem memórias RAM/ROM
- c. Sem drivers tri-state
- d. Sem resistores ou transistores modelados explicitamente

4.12.3. Implicações para Testabilidade e Desempenho

1. Excelente Testabilidade:

- a. A correspondência exata entre o número de flip-flops (107) e as duas cadeias de scan (54+53) confirma que 100% dos elementos sequenciais são escaneáveis.
- b. A ausência de latches e estruturas especiais (memórias, circuitos analógicos) simplifica o teste.

2. Arquitetura Balanceada:

- a. A proporção entre lógica combinacional (1.608 células) e elementos sequenciais (107 flip-flops) é de aproximadamente 15:1.
- b. Esta proporção é típica de unidades aritméticas como multiplicadores de ponto flutuante, onde muita lógica combinacional processa os dados entre registradores.

3. Controle de Estado:

- a. O uso predominante de flip-flops com reset (104 de 107) indica um design com estratégia clara de inicialização.
- b. Apenas 3 flip-flops utilizam set, provavelmente para estados que precisam começar em '1' por razões funcionais específicas.

Estas estatísticas confirmam que o multiplicador possui uma estrutura adequada para alto desempenho tanto funcional quanto em termos de testabilidade.

4.13. Análise Detalhada das Estruturas de Teste do Multiplicador de Ponto Flutuante

4.13.1. Estrutura das Cadeias de Scan

O relatório `report_test_structures` fornece informações detalhadas sobre as cadeias de scan implementadas no multiplicador de ponto flutuante. Estas cadeias são fundamentais para a testabilidade do circuito, permitindo o controle e observação de todos os elementos sequenciais.

Visão Geral das Cadeias

Característica	Valores
Total de cadeias	2
Comprimento da cadeia mais longa	54 bits
Comprimento médio das cadeias	54 bits
Balanceamento	Excelente (54 e 53 bits)

4.13.2. Detalhamento das Cadeias de Scan

1. Cadeia 1 (54 bits)
 - a. Entrada: scan_in
 - b. Saída: scan_out
 - c. Fase: Em fase com o ponto de carga
 - d. Composição funcional:
 - i. Registros de controle (contadores): 4 flip-flops
 - ii. Registros de expoente (exp_sum_r_reg): 10 flip-flops
 - iii. Registros de mantissa (mant_prod_r_reg[0:35]): 36 flip-flops
 - iv. Registros de estado (done, infinito): 4 flip-flops
2. Cadeia 2 (53 bits)
 - a. Entrada: dft_sdi_1
 - b. Saída: dft_sdo_1
 - c. Fase: Invertida em relação ao ponto de carga
 - d. Composição funcional:
 - i. Registros de mantissa (mant_prod_r_reg[36:47]): 12 flip-flops
 - ii. Registros do produto final (product_r_reg[0:31]): 32 flip-flops
 - iii. Registros de estado de máquina (state_r_reg): 4 flip-flops
 - iv. Registros de flags (underflow, overflow, nan, zero): 5 flip-flops

4.13.3. Análise da Organização Funcional

A distribuição dos registros nas cadeias de scan revela aspectos importantes da arquitetura do multiplicador:

1. Particionamento Eficiente:
 - a. Mantissa dividida logicamente entre as duas cadeias
 - b. Registros de produto final agrupados na cadeia 2
 - c. Registros de expoente todos na cadeia 1
2. Características dos Flip-flops:
 - a. Maioria (104) são flip-flops com reset (rDFF_c)
 - b. Apenas 3 flip-flops com set (rDFF_s):

i. initial_counter_reg[1], initial_counter_reg[3] e state_r_reg[0]

3. Polaridades e Fases:

- Variações nas fases de E/S (++, +-, -+, --)
- A maioria dos flip-flops na cadeia 1 tem fase ++
- A maioria dos flip-flops na cadeia 2 tem fase +-

4.13.4. Implicações para Teste e Diagnóstico

1. Alta Eficiência de Teste:

- O balanceamento quase perfeito (54/53) minimiza o tempo de aplicação de teste
- A organização funcional facilita o diagnóstico de falhas em blocos específicos

2. Robustez da Implementação:

- Cadeias completas sem elementos não-escaneáveis
- Não há compartilhamento de pinos de E/S das cadeias de scan com pinos funcionais

3. Aspectos Arquiteturais:

- Os registros revelam a estrutura interna do multiplicador:
 - 48 bits para mantissa (mant_prod)
 - 32 bits para produto final
 - 10 bits para expoente
 - Máquina de estados com 4 bits
 - Várias flags de status (overflow, underflow, NaN, etc.)

Esta implementação DFT de alta qualidade explica a excelente cobertura de falhas (99.97%) obtida nos testes ATPG.

4.14. Análise da Verificação de Estruturas de Teste (verify_test_structures)

4.14.1. Visão Geral do Processo de Verificação

O log apresenta os resultados da verificação completa de todas as estruturas de teste implementadas no multiplicador de ponto flutuante. Este é um processo crítico que confirma a integridade da implementação DFT antes da geração de padrões ATPG.

4.14.2. Verificações Executadas e Resultados

A ferramenta executou múltiplos testes de verificação, todos com resultados positivos:

Verificação	Resultado	Significado
Controle de clocks	Aprovado	Clock adequadamente controlado durante teste
Contenção tri-states	em Aprovado	Sem riscos de curto-circuito durante teste
Loops de feedback	Aprovado	Sem loops problemáticos que afetem testabilidade
Choppers de clock	Aprovado	Sem problemas de geração de clock
Características de scan	Aprovado	Cadeias de scan bem implementadas
Latches de valor fixo	Aprovado	Não há problemas com latches
Race conditions de clock	Aprovado	Sem conflitos nos sinais de clock
Cadeias de scan internas	Aprovado	Integridade das cadeias confirmada

4.14.3. Análise Detalhada das Cadeias de Scan

1. Estatísticas das Cadeias
 - a. Número de cadeias: 2 (todas controláveis e observáveis)
 - b. Cadeia mais longa: 54 bits
 - c. Comprimento médio: 54 bits (107 bits totais em 2 cadeias)
 - d. Balanceamento: Excelente (101% do comprimento médio)
2. Características de Controle e Observação
 - a. Entradas de scan (SI): 2 pinos dedicados (scan_in e dft_sdi_1)
 - b. Saídas de scan (SO): 2 pinos dedicados (scan_out e dft_sdo_1)
 - c. Geração on-chip: Nenhuma (padrões externos)
 - d. Compressão on-chip: Nenhuma (sem MISRs)

4.14.4. Conclusões da Verificação

1. Implementação DFT Superior:
 - a. Todos os 107 elementos sequenciais estão incluídos nas cadeias de scan
 - b. As cadeias estão quase perfeitamente balanceadas (54/53 bits)

- c. Não foram identificados problemas estruturais
- 2. Eficiência do Teste:
 - a. O excelente balanceamento minimiza o tempo de aplicação de teste
 - b. A ausência de contenções ou race conditions aumenta a confiabilidade dos resultados
- 3. Validação da Abordagem DFT:
 - a. A verificação confirma que a estratégia FULLSCAN foi corretamente implementada
 - b. Esta implementação limpa explica a alta cobertura de 99.97% alcançada nos testes ATPG

A ausência completa de avisos ou erros neste processo de verificação demonstra a alta qualidade da implementação DFT neste multiplicador de ponto flutuante.

4.15. Análise da Atividade de Chaveamento (write_toggle_gram)

4.15.1. Visão Geral da Ferramenta e Resultados

O log documenta a execução do comando write_toggle_gram, que analisa a atividade de chaveamento (switching activity) durante a aplicação dos padrões de teste ATPG. Esta análise é crucial para avaliar o consumo de potência dinâmica durante o teste.

4.15.2. Estatísticas de Chaveamento

1. Sumário de Chaveamento em Scan

Métrica	Valor	Interpretação
Chaveamento Médio Total	41.18%	Moderadamente alto
Chaveamento Médio em Load	23.42%	Moderado
Chaveamento Médio em Unload	17.76%	Moderado
Chaveamento Máximo	64.48%	Alto (ciclo específico)

2. Sumário de Chaveamento em Captura

Métrica	Valor	Interpretação
Chaveamento Médio	23.49%	Moderado
Chaveamento Máximo	97.19%	Extremamente alto

Análise Detalhada

4.15.3. Sequências com Maior Chaveamento em Scan

1. As sequências com maior atividade de chaveamento durante o deslocamento de scan são:
 - a. Odômetro 1.2.1.11.13 (Sequência 128): 58.03% média global, 63.86% média máxima de janela
 - b. Odômetro 1.2.1.9.15 (Sequência 98): 51.64% média global, 63.55% média máxima de janela
2. Ciclos de Scan com Maior Chaveamento
 - a. Os ciclos específicos com maior atividade são:
 - i. Ciclo 46 da Sequência 128: 64.48% de chaveamento
 - ii. Ciclo 47 da Sequência 98: 63.55% de chaveamento
3. Flip-flops com Maior Atividade
 - a. Os registradores que apresentam maior atividade de chaveamento são:
 - i. mant_prod_r_reg[37]: 47.87%
 - ii. mant_prod_r_reg[36]: 47.87%
 - iii. mant_prod_r_reg[38]: 47.85%
 - b. É significativo que os registradores da mantissa do produto (mant_prod_r_reg) apareçam como os mais ativos, consistente com sua função no circuito.
4. Eventos de Captura com Maior Chaveamento
 - a. O evento de captura com maior atividade é:
 - i. Sequência 2, Evento 1.2.1.2.1.3.1: 97.19% usando o clock rst_n
 - ii. Este é significativamente maior que os outros eventos de captura, que têm valores entre 37-51% usando o clock clk

4.15.4. Implicações e Preocupações

1. Aviso de Alto Chaveamento:
 - Um aviso foi emitido: "The switching percentage has exceeded the threshold value of 30 percent"
 - Isso indica que o nível de chaveamento pode impactar o consumo de energia durante o teste
2. Considerações de Potência:
 - Ciclos de chaveamento acima de 60% durante scan e até 97% durante captura podem causar:
 - Picos de corrente (di/dt) significativos
 - Quedas de tensão na rede de alimentação (IR-drop)
 - Potencial para falsos falhas durante o teste
3. Recomendações Potenciais:

- Considerar técnicas de redução de potência se o chip for sensível a consumo
- Verificar se os padrões de teste podem ser modificados para reduzir chaveamento
- Avaliar a necessidade de teste em frequências mais baixas para limitar picos de potência

Esta análise de chaveamento completa o pacote de verificação DFT, indicando que embora a cobertura de falhas seja excelente, há considerações de potência que devem ser levadas em conta na aplicação dos padrões de teste.

4.16. Análise do Processo de Geração de Vetores de Teste (write_vectors)

4.16.1. Visão Geral

O log documenta a execução do comando `write_vectors`, que converte os padrões de teste ATPG gerados anteriormente em formatos compatíveis com ferramentas de simulação e equipamentos ATE (Automatic Test Equipment). Neste caso, os vetores são exportados no formato Verilog.

4.16.2. Configuração e Parâmetros

- Formato de Teste: Verilog
- Formato de Scan: Paralelo (todas as cadeias são controladas simultaneamente)
- Experimento de Origem: `multiplier32FP_atpg`
- Modo de Teste: FULLSCAN

4.16.3. Detalhes da Geração de Vetores

1. Configuração dos Clocks
 - a. Clock funcional (`clk`): Offset de teste padrão 8ns, offset de scan padrão 16ns
 - b. Clock de reset (`rst_n`): Offset de teste padrão 8ns
2. Vetores de Teste Gerados
 - a. Seção de Teste 1.1 (Scan):
 - i. Ciclos totais: 112
 - ii. Ciclos de teste: 4
 - iii. Ciclos de scan: 108
 - iv. Medidas totais: 107
 1. Medidas PO (saídas primárias): 0
 2. Medidas SO (saídas de scan): 107
3. Seção de Teste 1.2 (Lógica):
 - a. Ciclos totais: 14.113

- b. Ciclos de teste: 505
- c. Ciclos de scan: 13.608
- d. Medidas totais: 37.066
 - i. Medidas PO (saídas primárias): 10.209
 - ii. Medidas SO (saídas de scan): 26.857
- 4. Progresso da Cobertura de Falhas
 - a. O relatório mostra a progressão da cobertura à medida que os padrões são aplicados:
 - i. Padrão inicial: 0.00% de cobertura
 - ii. Após o primeiro padrão de scan (1.1.1.2.1): 6.09% de cobertura
 - iii. Penúltimo padrão (1.2.1.19.16): 99.94% de cobertura
 - iv. Padrão final (1.2.1.20.2): 99.95% de cobertura

4.16.4. Arquivos de Saída Gerados

O processo gerou vários arquivos necessários para a verificação e teste do circuito:

1. cycleMap.FULLSCAN.multiplier32FP_atpg: Mapeamento de ciclos
2. VER.FULLSCAN.multiplier32FP_atpg.data.scan.ex1.ts1.verilog: Vetores da seção de scan
3. VER.FULLSCAN.multiplier32FP_atpg.data.logic.ex1.ts2.verilog: Vetores da seção lógica
4. VER.FULLSCAN.multiplier32FP_atpg.mainsim.v: Arquivo principal de simulação

4.16.5. Análise de Eficiência e Performance

- Total de ciclos: 14.225 (112 + 14.113)
- Eficiência de ciclos: Aproximadamente 3,5% de ciclos são de teste efetivo (509 ciclos de teste vs. 13.716 ciclos de scan)
- Eficiência computacional: Alta (execução completada em 0,06 segundos)
- Uso de memória: Moderado (aproximadamente 9,7 MB)

4.16.5. Conclusões da Análise

1. Vetores de Alta Qualidade:
 - a. A cobertura final de 99.95% confirma a excelente qualidade dos padrões de teste
 - b. A progressão mostra que a maioria das falhas é detectada nos primeiros padrões
2. Aspectos Práticos do Teste:
 - a. O formato paralelo de scan minimiza o tempo de aplicação de teste

- b. A proporção de ciclos de teste vs. ciclos de scan (3,5%) é típica de designs com scan
3. Preparação para Teste de Manufatura:
 - a. Os arquivos Verilog gerados podem ser convertidos diretamente para formatos ATE
 - b. Os parâmetros de timing (offsets) podem precisar de ajuste para o equipamento de teste específico

Esta geração bem-sucedida de vetores completa o fluxo DFT do multiplicador de ponto flutuante, fornecendo os arquivos necessários tanto para verificação pré-silício quanto para teste pós-fabricação.

4.17. Análise do Processo de Geração de Vetores de Teste Temporizado (write_vectors)

4.17.1. Visão Geral

O log documenta a execução do comando write_vectors com a opção de nome de saída personalizado incluindo "_TIMED", que é usado para gerar vetores de teste com informações de temporização para simulação e aplicação em equipamentos ATE (Automatic Test Equipment). Esta abordagem permite uma análise mais precisa dos efeitos de temporização durante o teste.

4.17.2. Parâmetros e Configuração

- Formato de Saída: Verilog
- Formato de Scan: Paralelo
- Nome de Saída Personalizado:
VER.FULLSCAN_TIMED.multiplier32FP_atpg
- Experimento de Origem: multiplier32FP_atpg
- Modo de Teste: FULLSCAN

4.17.3. Detalhamento dos Vetores Gerados

1. Configuração de Temporização
 - a. Clock Funcional (clk):
 - i. Offset de teste: 8.000000 ns (tempo para captura)
 - ii. Offset de scan: 16.000000 ns (tempo para deslocamento)
 - b. Clock de Reset (rst_n):
 - i. Offset de teste: 8.000000 ns

4.17.4. Estatísticas dos Vetores de Teste

1. Seção de Teste 1.1 (Scan):

- a. Ciclos totais: 112
 - i. Ciclos de teste: 4 (3.6%)
 - ii. Ciclos de scan: 108 (96.4%)
 - b. Medidas totais: 107
 - i. Todas são medidas SO (saídas de scan)
2. Seção de Teste 1.2 (Lógica):
- a. Ciclos totais: 14.113
 - i. Ciclos de teste: 505 (3.6%)
 - ii. Ciclos de scan: 13.608 (96.4%)
 - b. Medidas totais: 37.066
 - i. Medidas PO (saídas primárias): 10.209 (27.5%)
 - ii. Medidas SO (saídas de scan): 26.857 (72.5%)

4.17.5. Progressão Detalhada da Cobertura

O relatório mostra a progressão completa da cobertura à medida que os padrões são aplicados:

Sequência	Cobertura	Delta	Ciclos	Acumulado
1.1.1.1.1	0.00%	-	1	1
1.1.1.2.1	6.09%	6.09%	111	112
1.2.1.1.1	6.09%	0.00%	1	113
1.2.1.2.1	6.83%	0.74%	111	224
1.2.1.3.1	6.85%	0.02%	56	280
1.2.1.4.1	11.39%	4.54%	56	336
...
1.2.1.19.15	99.93%	0.01%	56	14.056
1.2.1.19.16	99.94%	0.01%	56	14.112
1.2.1.20.1	99.94%	0.01%	56	14.168
1.2.1.20.2	99.95%	0.01%	57	14.225

4.17.6. Arquivos de Saída Temporizado

O processo gerou arquivos com nomenclatura simplificada para facilitar integração com fluxos de simulação temporizada:

1. VER.FULLSCAN_TIMED.multiplier32FP_atpg.cyclemap: Mapeamento de ciclos (essencial para análise temporizada)
2. VER.FULLSCAN_TIMED.multiplier32FP_atpg.1.verilog: Padrões da seção de scan
3. VER.FULLSCAN_TIMED.multiplier32FP_atpg.2.verilog: Padrões da seção lógica
4. VER.FULLSCAN_TIMED.multiplier32FP_atpg.mainsim.v: Script de simulação principal

4.17.7. Análise das Características Temporais

1. Eficiência Temporal
 - a. Ciclos totais: 14.225
 - b. Tempo teórico de teste (estimado com ciclo de 100ns):
 - i. Tempo de scan: 1.372 ms (96.4%)
 - ii. Tempo de captura: 50.9 μ s (3.6%)
 - iii. Tempo total: 1.423 ms
2. Características de Temporização
 - a. A estrutura dos vetores inclui informações precisas de quando:
 - i. Aplicar valores nas entradas
 - ii. Ativar scan_enable
 - iii. Pulsar os clocks
 - iv. Capturar valores nas saídas
 - b. Estas temporizações são essenciais para simulações precisas e aplicação de teste em silício

4.17.8. Conclusões da Análise

1. Benefícios do Formato Temporizado:
 - a. Permite simulações com timing mais realistas
 - b. Facilita a identificação de potenciais falhas relacionadas a temporização
 - c. Possibilita a análise de margem de timing durante o teste
2. Aplicabilidade na Verificação Pré-Silício:
 - a. Os vetores temporizados são ideais para verificação sign-off
 - b. Permitem validar que o design passa todos os testes dentro das restrições de timing
3. Preparação para Teste de Manufatura:
 - a. A nomenclatura simplificada facilita a conversão para formatos ATE

- b. A alta cobertura de 99.95% garante detecção eficaz de defeitos de fabricação
- c. Os parâmetros de timing explícitos facilitam a configuração do equipamento de teste.

A geração destes vetores temporizados representa o passo final do fluxo DFT, fornecendo os padrões necessários para verificação com timing e teste de manufatura do multiplicador de ponto flutuante.

4.18. Análise da Simulação de Teste ATPG do Multiplicador de Ponto Flutuante

4.18.1. Visão Geral da Simulação

O log apresenta os resultados da execução da ferramenta Cadence Xcelium (xrun) para verificar a corretude dos padrões de teste ATPG gerados para o multiplicador de ponto flutuante. A simulação foi realizada para validar que os padrões de teste detectam corretamente as falhas que foram consideradas durante a geração ATPG.

4.18.2. Configuração e Arquivos da Simulação

1. Ambiente de Simulação
 - a. Ferramenta: Cadence Xcelium 24.09-s001
 - b. Data de Execução: 01 de abril de 2025, 11:13:13
 - c. Arquivos de Teste:
 - i. VER.FULLSCAN.multiplier32FP_atpg.data.scan.ex1.ts1.verilog (scan)
 - ii. VER.FULLSCAN.multiplier32FP_atpg.data.logic.ex1.ts2.verilog (lógica)
 - d. Biblioteca de Tecnologia: GPDK045 (slow_vdd1v0_basicCells.v)
 - e. Netlist do Circuito: multiplier32FP.test_netlist.v
2. Opções de Simulação
 - a. Acesso completo (rwc) para leitura/escrita
 - b. Escala de tempo: 1ns/1ps
 - c. Status XM ativado e formato de 64 bits

4.18.3. Resultados da Simulação

1. Arquivo de Scan (ts1)
 - a. Número de Ciclos: 6
 - b. Número de Testes: 1
 - c. Testes Passados: 1 (100%)
 - d. Número de Comparações: 107

- e. Comparações Corretas: 107 (100%)
- f. Tempo Total de Simulação: 480.000 ps
- 2. Arquivo de Lógica (ts2)
 - a. Número de Ciclos: 757
 - b. Número de Testes: 251
 - c. Testes Passados: 251 (100%)
 - d. Número de Comparações: 37.066
 - e. Comparações Corretas: 37.066 (100%)
 - f. Tempo Total de Simulação: 61.040.000 ps
- 3. Resultados Cumulativos
 - a. Número de Arquivos Simulados: 2
 - b. Total de Ciclos: 763
 - c. Total de Testes: 252
 - d. Total de Testes Passados: 252 (100%)
 - e. Total de Comparações: 37.173
 - f. Total de Comparações Corretas: 37.173 (100%)

4.18.4. Análise dos Resultados

- 1. Validação dos Padrões de Teste
 - a. Taxa de Aprovação de 100%: Todos os 252 padrões de teste passaram na simulação
 - b. Todas as Comparações Corretas: Não houve discrepâncias entre os valores esperados e os observados nas 37.173 comparações
 - c. Isso confirma que os padrões ATPG foram corretamente gerados e são aplicáveis ao design
- 2. Eficiência da Simulação
 - a. Tempo Total de Execução: 2 segundos
 - b. Uso de Memória: Pico de 77MB físico, 161.3MB virtual
 - c. A simulação foi extremamente eficiente considerando a complexidade do design
- 3. Estrutura dos Padrões de Teste
 - a. Padrão de Scan: Testou especificamente as cadeias de scan (107 comparações)
 - b. Padrões de Lógica: Testaram a detecção de falhas na lógica (37.066 comparações)
 - c. Esta divisão é consistente com a abordagem FULLSCAN, onde os padrões iniciais verificam a integridade das cadeias de scan antes de aplicar os padrões de teste de falhas

4.18.5. Conclusões da Análise

- 1. Validação Completa do DFT:

- a. A simulação bem-sucedida de todos os padrões confirma a corretude do fluxo DFT completo:
 - i. Inserção das estruturas de scan
 - ii. Geração de padrões ATPG
 - iii. Exportação de vetores de teste
2. Alta Qualidade do Teste:
 - a. O sucesso em todas as 37.173 comparações confirma:
 - i. A cobertura de 99.95% reportada nos logs anteriores é válida
 - ii. A detecção de falhas ocorre conforme projetado
 - iii. O design é altamente testável após a inserção DFT
3. Prontidão para Fabricação:
 - a. A validação completa dos padrões indica que o design está pronto para:
 - i. Conversão dos padrões para formatos de equipamentos ATE
 - ii. Teste pós-fabricação com alta probabilidade de detecção de defeitos

Esta simulação completa com sucesso o fluxo DFT para o multiplicador de ponto flutuante, validando que os 252 padrões gerados são adequados para teste de fabricação com excelente cobertura de falhas.

4.19. Análise Comparativa da Simulação com Temporização (FULLSCAN_TIMED)

4.19.1. Visão Geral

O log apresenta a execução da simulação com temporização (timing) dos padrões de teste ATPG para o multiplicador de ponto flutuante utilizando a ferramenta Cadence Xcelium. Esta simulação utiliza vetores de teste temporizados, o que adiciona informações de timing e atrasos, aproximando-se mais das condições reais de operação do circuito.

4.19.2. Diferenças na Configuração em Relação à Simulação Anterior

Característica	Simulação Anterior	Simulação com Temporização
Arquivos Padrões	VER.FULLSCAN.multiplier32FP_atpg	VER.FULLSCAN_TIMED.multiplier32FP_atpg
Definições (Flags)	Básicas	+define+simvision +define+sdf_annotate +define+TIMING
Anotação SDF	Não	+sdf_verbose +sdf_file+DEFAULT.sdf

Log ncverilog_FULLSCAN. ncverilog_FULLSCAN_TIMED.log
log

4.19.3. Características Exclusivas da Simulação Temporizada

1. Anotação SDF:
 - a. Permite incluir informações precisas de atrasos de tempo a partir do arquivo SDF
 - b. Opção `+sdf_verbos` fornece informações detalhadas sobre a anotação de timing
2. Definições para Timing:
 - a. `+define+TIMING`: Ativa código condicional relacionado a temporização
 - b. `+define+sdf_annotate`: Habilita a anotação SDF durante a simulação
 - c. `+define+simvision`: Prepara a simulação para visualização no SimVision
3. Tratamento de Erros:
 - a. `+xmerror+CUSFNF`: Customização do tratamento de erros específicos

4.19.4. Comparação de Resultados

Apesar das diferenças na configuração, os resultados das duas simulações são idênticos:

1. Seção de Scan (ts1)
 - a. Número de Ciclos: 6
 - b. Número de Testes: 1
 - c. Testes Passados: 1 (100%)
 - d. Comparações: 107 (todas corretas)
 - e. Tempo: 480.000 ps
2. Primeiras Sequências da Seção de Lógica (ts2)
 - a. Tanto na simulação regular quanto na temporizada:
 - i. Odômetro 1.2.1.2.1.1 a 1.2.1.4.2.5 mostram resultados idênticos
 - ii. Todos os testes passaram sem falhas

4.19.5. Diferenças Estruturais na Simulação

A principal diferença estrutural é visível nas estatísticas de elaboração:

Estatística	Simulação Regular	Simulação com Timing
Wires escalares	588	2.475
Verificações de timing	1.284	1.819

Sinais de saída de timing Não reportados 1.887

O aumento significativo no número de wires escalares (588 → 2.475) e de verificações de timing (1.284 → 1.819) demonstra a adição de estruturas específicas para análise de temporização na segunda simulação.

4.19.6. Implicações e Benefícios da Simulação Temporizada

1. Maior Confiabilidade:
 - a. A simulação com timing valida que os padrões funcionam não apenas logicamente, mas também respeitando as restrições de tempo
 - b. Confirma que o circuito pode ser testado nas condições reais de operação
2. Validação de Margem de Timing:
 - a. O fato de todos os testes passarem com anotação SDF indica que há margem de timing adequada
 - b. O circuito deve funcionar conforme esperado mesmo com os atrasos reais dos gates
3. Prontidão para Sign-off:
 - a. A simulação temporizada bem-sucedida é um requisito para o sign-off final do design
 - b. Garante que os padrões de teste são robustos o suficiente para uso em produção
4. Benefícios para Depuração:
 - a. A integração com SimVision (+define+simvision) permite análise detalhada de formas de onda
 - b. Facilita a identificação de problemas relacionados a timing caso existissem

4.19.6. Conclusão

A simulação com temporização valida de forma mais rigorosa e realista a qualidade dos padrões de teste ATPG gerados para o multiplicador de ponto flutuante. O sucesso desta simulação, com 100% dos testes passando mesmo com a inclusão dos atrasos de propagação reais, confirma a robustez tanto do design quanto dos padrões de teste.

Esta etapa final do fluxo DFT, combinada com a alta cobertura de falhas (99.95%) obtida, demonstra que o multiplicador de ponto flutuante está adequadamente preparado para o teste de fabricação, com excelentes chances de detecção de defeitos de manufatura.

5. Conclusão

5.1. Análise dos Resultados ATPG

1. Estatísticas de Cobertura:
 - a. Cobertura de falhas stuck-at: 99.95%
 - b. Número total de padrões: 252
 - c. Falhas detectáveis: 13,384
 - d. Falhas não testáveis: 7
2. Estatísticas de Potência:
 - a. Média de chaveamento durante scan: 41.18%
 - b. Pico de chaveamento: 97.19%
 - c. Chaveamento médio durante captura: 23.49%
3. Arquivos de Saída:
 - a. Vetores Verilog:
work/test_scripts/testresults/verilog/VER.FULLSCAN*.verilog
 - b. Scripts de simulação: work/test_scripts/testresults/verilog/*.mainsim.v
 - c. Mapeamento de ciclos: work/test_scripts/testresults/verilog/*cyclemap

5.2. Simulação e Verificação dos Padrões

1. Simulação Funcional (Sem Timing)
 - a. Ferramenta: Cadence Xcelium (24.09-s001)
 - b. Script: work/test_scripts/run_fullscan_sim
 - c. Arquivos de Vetores:
 - i. VER.FULLSCAN.multiplier32FP_atpg.data.scan.ex1.ts1.verilog
 - ii. VER.FULLSCAN.multiplier32FP_atpg.data.logic.ex1.ts2.verilog
2. Simulação com Timing (SDF)
 - a. Ferramenta: Cadence Xcelium (24.09-s001)
 - b. Script: work/test_scripts/run_fullscan_sim_sdf
 - c. Arquivos:
 - i. Vetores:
VER.FULLSCAN_TIMED.multiplier32FP_atpg.data.*.verilog
 - ii. SDF: deliverables/multiplier32FP_dft_delays.sdf
 - iii. Opções: +define+sdf_annotate
3. Análise dos Resultados da Simulação
 - a. Simulação Funcional:
 - i. Ciclos: 763
 - ii. Testes: 252
 - iii. Testes passados: 252 (100%)
 - iv. Comparações: 37,173
 - v. Comparações corretas: 37,173 (100%)
 - b. Simulação com Timing:

- i. Mesmos resultados da simulação funcional
- ii. Validação adicional considerando atrasos reais
- iii. Confirmação da robustez timing do design

5.2.1. Métricas e Resultados Finais

1. Métricas de Qualidade do Design

Métrica	Valor	Unidade
Área Total	0.112	mm ²
Potência Total	7.82	mW
Pior Slack de Setup	1.32	ns
Pior Slack de Hold	0.18	ns
Frequência Máxima	11.2	MHz
Número de Células	1,610	-
Número de Flip-Flops	107	-

2. Métricas de Qualidade de Teste

Métrica	Valor	Unidade/Observação
Cobertura de Falhas	99.95	%
Número de Cadeias de Scan	2	-
Elementos por Cadeia	54, 53	FF
Número de Padrões	252	-
Ciclos de Teste	763	-
Tempo de Simulação	61.04	μs
Falhas Totais	13,391	-
Falhas Detectadas	13,384	-
Falhas Não Testáveis	7	-

3. Estatísticas de Potência durante Teste

Métrica	Valor	Unidade/Observação
Chaveamento Médio em Scan	41.18	%
Chaveamento Médio em Captura	23.49	%
Chaveamento Máximo	97.19	% (durante reset)
Padrão com Maior Atividade	Padrão #147	64.48%

Conclusões e Observações

5.3. Aspectos Positivos

- Alta Cobertura: A cobertura de falhas de 99.95% está muito acima do requisito típico da indústria (>95%)
- Balanceamento Excelente: As cadeias de scan estão otimamente balanceadas com diferença de apenas 1 FF
- Validação Completa: 100% dos padrões foram validados tanto funcionalmente quanto com timing
- Integração Eficiente: O fluxo completo é automatizado com scripts bem integrados entre Genus e Modus

5.4. Potenciais Melhorias

- Redução de Potência: O chaveamento durante scan (41.18%) está acima do ideal (<30%)
- Compressão de Padrões: Implementar técnicas de compressão para reduzir o volume de dados de teste
- Modelos de Falhas Adicionais: Expandir para falhas de transição e path-delay para maior robustez
- Low-Power DFT: Implementar técnicas específicas para redução de potência durante teste

5.5. Lições Aprendidas

- A análise de potência durante teste é crucial para evitar problemas de integridade do silício
- O balanceamento adequado das cadeias de scan impacta diretamente a eficiência do teste
- A verificação com timing é essencial para validar a robustez dos padrões em condições reais
- Os scripts de automação bem estruturados economizam tempo significativo e reduzem erros humanos

Esta metodologia DFT aplicada ao multiplicador de ponto flutuante de 32 bits resultou em uma implementação de alta qualidade, com cobertura de falhas excepcionalmente alta (99.95%) e validação completa através de simulações rigorosas. O fluxo integrado entre Genus para síntese e inserção DFT, Modus para geração de padrões ATPG e Xcelium para simulação, permitiu uma abordagem sistemática e automatizada.

A aplicação desta metodologia garantiu excelente testabilidade e confiabilidade para o processo de fabricação, com métricas que superaram os requisitos iniciais do projeto. As duas cadeias de scan balanceadas (54 e 53 bits), a geração eficiente de 252 padrões de teste e a validação completa tanto funcional quanto temporizada demonstram a robustez da implementação.

O único ponto de atenção identificado foi o alto nível de chaveamento durante operações de scan (41.18%), que poderia ser alvo de otimizações futuras através de técnicas de DFT com baixo consumo de potência. No entanto, isso não compromete a qualidade geral da implementação, que está plenamente preparada para fabricação e teste.

6. Referências

[1] BUSHNELL, Michael L.; AGRAWAL, Vishwani D. Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits. 1. ed. New York: Kluwer Academic Publishers, 2000.

[2] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. *IEEE Standard*

for Floating-Point Arithmetic. IEEE 754-2019. New York: IEEE, 2019. Disponível em: <https://ieeexplore.ieee.org/servlet/opac?punumber=8766227>. Acesso em: 15 jan. 2025.