

DISSERTAÇÃO DE MESTRADO Nº 1036

**AUMENTO DE VIDA ÚTIL DE CIRCUITOS INTEGRADOS DIGITAIS COM  
TÉCNICAS DE DETECÇÃO E CORREÇÃO DE ERROS TEMPORAIS IN SITU**

**Andrei dos Santos Silva**

DATA DA DEFESA: 09/02/2018



**Universidade Federal de Minas Gerais**

**Escola de Engenharia**

**Programa de Pós-Graduação em Engenharia Elétrica**

**AUMENTO DE VIDA ÚTIL DE CIRCUITOS INTEGRADOS  
DIGITAIS COM TÉCNICAS DE DETECÇÃO E CORREÇÃO DE  
ERROS TEMPORAIS IN SITU**

**Andrei dos Santos Silva**

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica.

**Orientador: Prof. Frank Sill Torres**

**Belo Horizonte - MG**

**Fevereiro de 2018**



**"Aumento de Vida Útil de Circuitos Integrados Digitais com Técnicas de Detecção e Correção de Erros Temporais *In Situ*"**

**Andrei dos Santos Silva**

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

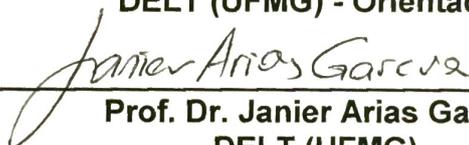
Aprovada em 09 de fevereiro de 2018.

Por:



---

**Prof. Dr. Frank Sill Torres**  
DELT (UFMG) - Orientador



---

**Prof. Dr. Janier Arias García**  
DELT (UFMG)



---

**Prof. Dr. Omar Paranaíba Vilela Neto**  
DCC (UFMG)



Andrei dos Santos Silva

**Aumento de Vida Útil de Circuitos Integrados  
Digitais com Técnicas de Detecção e Correção  
de Erros Temporais In Situ**

**Brasil**

**Fevereiro de 2018**



Andrei dos Santos Silva

**Aumento de Vida Útil de Circuitos Integrados Digitais  
com Técnicas de Detecção e Correção de Erros  
Temporais In Situ**

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica.

Universidade Federal de Minas Gerais – UFMG

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

Brasil

Fevereiro de 2018



# Agradecimentos

Gostaria de agradecer à minha mãe, Lisiane, por me incentivar a dar sempre o melhor de mim, uma companheira para todos os momentos, inclusive os mais difíceis. À minha irmã, Isabela, por conceder-me a paz necessária para trabalhar. Ao meu pai, Manoel, por apoiar incondicionalmente esse empreendimento.

Ao meu orientador, Prof. Frank Sill Torres, o melhor mentor acadêmico que um aluno poderia ter. Obrigado pela paciência e pelas horas desprendidas de trabalho.

Aos professores Omar Paranaíba Vilela Neto e Janier Arias García pela disponibilidade e paciência para participar da banca avaliadora deste trabalho.

A todos os meus colegas e amigos do laboratório OptMA por me aturarem ao longo dessa jornada.



Eu relembro amigos de guerras quase esquecidas,  
Todos eles repartidos em cada ferimento recebido,  
Ferimentos que lembram lugares dolorosos onde lutamos,  
Batalhas que é melhor esquecer, coisas que nunca buscamos,  
Que foi que perdemos e que conseguimos?

- Canções da Dispersão  
(Hereges de Duna, Frank Herbert)



# Resumo

Avanços nos processos de fabricação de dispositivos microeletrônicos integrados levaram à miniaturização dos transistores de efeito de campo (FETs) à escala nanométrica. O tamanho reduzido desses elementos beneficiou os sistemas computacionais com o aumento da velocidade de operação dos microprocessadores e possibilitou uma maior integração de funcionalidades.

À medida que esses avanços entraram na região submicron, no entanto, os dispositivos microeletrônicos passaram a apresentar problemas de confiabilidade. Ruídos eletromagnéticos, variações oriundas dos processos de fabricação e efeitos de degradação foram identificados como os principais agentes no aumento da susceptibilidade de circuitos integrados à falhas físicas. Na literatura, muitos trabalhos focam no aumento da confiabilidade dos circuitos integrados através da inserção de elementos sequenciais capazes de detectar e corrigir erros *in situ*. Esses trabalhos, no entanto, apresentam técnicas que dependem da suspensão da máquina ou do *clock* de operação para realizar a recuperação de erros.

O presente trabalho apresenta duas técnicas de detecção e correção de erros em circuitos sequenciais complexos que não exigem interrupção de operação. A primeira é o Empréstimo Seletivo de Tempo (*Selective Time-borrowing*), caracterizada pela realocação de tempo dos estágios menos congestionados do circuito para os caminhos lógicos mais críticos. A segunda foi desenvolvida para quando o empréstimo temporal não é possível, chamada Ativação de Caminho Alternativo (*Alternative Path Activation*). Essa técnica cria um processamento paralelo para recuperar o erro sem que tempo seja consumido de outros estágios sequenciais da máquina. Ambas são voltadas para o aumento do tempo de vida de circuitos integrados digitais ao focar primariamente nos efeitos de degradação. É proposto, também, um fluxo de inserção automática dessas técnicas para ser utilizado durante o projeto de circuitos integrados.

A validação das técnicas foram efetuadas com simulações em nível de transistores. Um processador ARM foi modificado com as técnicas propostas e enviado para fabricação em tecnologia de 130nm. Avaliações acerca da aplicabilidade das técnicas também foram feitas utilizando vários circuitos de teste do ISPD12.

As técnicas propostas oferecem a possibilidade de aumentar a vida útil de circuitos integrados e prover maior confiabilidade na era da computação ubíqua.

**Palavras-chave:** confiabilidade. circuitos integrados digitais. vida útil. erros temporais.



# Abstract

Advances in the manufacturing processes of microelectronics devices led to an increase miniaturization of the Field Effect Transistors (FETs), reaching nanometer scale nowadays. The reduced size of such elements brought significant increase to performance and functionality integration of computational systems.

As these advances reach the submicron region, however, microelectronic devices begin to experience reliability issues during operation. Electromagnetic noise, fabrication process variations and aging effects have being identified as the main agents of integrated circuits increase susceptibility to physical failure. Many works in the literature focus in the reliability increase through the insertion of special sequential elements capable of *in situ* error detection and correction. These works, however, present techniques that require processing stall or clock frequency control in order to recover from such errors.

The current work presents two techniques of error detection and correction for complex sequential circuits that do not require machine stalling to operate. The first one is called Selective Time-Borrowing (EST). This technique operates through relocation of time between less constrained sequential stages in order to increase time margins for critical paths. The second one is directed to cases where no time margin is available for relocation, called Alternative Path Activation (APA). The APA uses parallel processing to recover from error without resorting to time-borrowing from other sequential stages. Both these techniques are directed at increasing digital integrated circuits reliability by targeting primarily aging effects. An automatic insertion flow to these techniques is proposed to be used on integrated circuits design.

In order to validate the proposed techniques, transistor and logic level simulations were conducted with test circuits and ISPD12 benchmarks. The logic level simulations focus on insertion of the techniques on more complex circuits, assessing coverage and time margin gain. An ARMv2 processor was modified manufactured in 130nm technology in order to test both techniques in a real circuit.

The proposed techniques offer the possibility to increase integrated circuits lifetime and reliability in the age of ubiquitous computing.

**Keywords:** reliability. integrated digital circuits. lifetime. timing errors.



# Lista de ilustrações

Figura 2.1 – Efeito da eletromigração em uma trilha metálica . . . . .	30
Figura 2.2 – Sistema RMN . . . . .	35
Figura 2.3 – Redundância ativa dupla . . . . .	36
Figura 2.4 – Redundância Híbrida . . . . .	36
Figura 2.5 – Diagrama do latch tipo D . . . . .	39
Figura 2.6 – Flip-flop tipo D . . . . .	40
Figura 2.7 – Diagramas temporais . . . . .	40
Figura 2.8 – Esquemático de um flip-flop . . . . .	40
Figura 2.9 – Diagrama de temporização real . . . . .	41
Figura 2.10–Diagrama temporal de um circuito com flip-flops . . . . .	43
Figura 2.11–Diagrama temporal de um circuito com duas fases . . . . .	43
Figura 2.12–Diagrama com <i>pipeline</i> e temporização dos dados . . . . .	44
Figura 2.13–Estados do latch . . . . .	45
Figura 2.14–Atraso gerado pela metaestabilidade . . . . .	46
Figura 2.15–Fluxo de projeto de circuitos integrados digitais . . . . .	47
Figura 2.16–Diagrama do flip-flop RAZOR . . . . .	50
Figura 2.17–Diagrama temporal da técnica RAZOR . . . . .	51
Figura 2.18–Flip-flop RAZOR com detector de metaestabilidade . . . . .	52
Figura 2.19–Diagrama funcional do RAZORII . . . . .	52
Figura 2.20–Diagrama temporal do RAZORII . . . . .	53
Figura 2.21–Diagrama funcional do SafeRAZOR . . . . .	55
Figura 3.1 – Diagrama temporal de detecção de erro . . . . .	59
Figura 3.2 – Circuito em <i>pipeline</i> operando em frequência máxima . . . . .	60
Figura 3.3 – <i>Pipeline</i> com EST . . . . .	62
Figura 3.4 – Empréstimo temporal utilizando atraso de cadeia de inversores . . . . .	63
Figura 3.5 – Diagrama de funcionamento do APA . . . . .	65
Figura 3.6 – Domínio duplicado (setas vermelhas) com envoltória de multiplexadores (verde) . . . . .	67
Figura 3.7 – Colisão entre domínios duplicados . . . . .	68
Figura 3.8 – Soluções para falso erro gerado por caminhos não críticos . . . . .	69
Figura 3.9 – Caminho crítico originando do domínio duplicado . . . . .	70
Figura 3.10–Diagrama do APAFF . . . . .	71
Figura 3.11–Detector de metaestabilidade . . . . .	72
Figura 4.1 – Diagramas utilizados em simulação . . . . .	76
Figura 4.2 – APAFF simulado no software MMSIM . . . . .	76

Figura 4.3 – Diagrama do circuito INVCHAIN com diferentes valores de inversores para os dois circuitos . . . . .	77
Figura 4.4 – Circuitos INVCHAIN modificados pelas técnicas . . . . .	78
Figura 4.5 – Diagrama funcional do compressor 4:2 . . . . .	78
Figura 4.6 – Somador do tipo <i>ripple-carry</i> . . . . .	79
Figura 4.7 – Circuito COMPRESSOR utilizado em simulação . . . . .	79
Figura 4.8 – Resposta do circuito INVCHAIN1 simulado . . . . .	81
Figura 4.9 – Gráfico com resultados da tabela 4 . . . . .	82
Figura 4.10–Gráfico com dados da Tabela 5 . . . . .	83
Figura 4.11–Gráfico com resultados da Tabela 6 . . . . .	85
Figura 4.12–Resposta do circuito INVCHAIN2 com APA . . . . .	85
Figura 4.13–Esquemático do flip-flop central simulado . . . . .	88
Figura 4.14–Situações de metaestabilidade observadas no APAFF . . . . .	89
Figura 4.15–Valores de ganho das janelas de detecção do APA com margem 75 % de CC e <i>mapa</i> 70 % . . . . .	93
Figura 4.16–Valores de ganho das janelas de detecção do APA para 90 % de margem e <i>mapa</i> 70 % . . . . .	94
Figura 4.17–Valores de ganho das janelas de detecção do APA para 90 % de margem e <i>mapa</i> . . . . .	95
Figura 4.18–Layout do microprocessador ARMv2-AMBER modificado pelo APA e EST . . . . .	98
Figura 4.19–Diagrama do processador AMBER . . . . .	99

# Lista de tabelas

Tabela 1 – Avanço tecnológico em processadores . . . . .	21
Tabela 2 – Tabela verdade do latch . . . . .	39
Tabela 3 – Tabela verdade do flip-flop sensível à borda de subida . . . . .	39
Tabela 4 – Resultados obtidos aplicando-se o EST ao INVCHAIN1 . . . . .	82
Tabela 5 – Resultados obtidos do EST no INVCHAIN2 . . . . .	83
Tabela 6 – Resultados obtidos de simulação com APA aplicado ao INVCHAIN2 . . . . .	84
Tabela 7 – Resultados das simulações com EST aplicado ao CC1 do COMPRESSOR . . . . .	86
Tabela 8 – Circuitos utilizados para teste de cobertura com APA e EST . . . . .	91
Tabela 9 – Resultados para inserção do APA e EST com 75 % de margem de CC e <i>mapa</i> 70 % . . . . .	92
Tabela 10 – Resultados para inserção do APA e EST com 90 % de margem e <i>mapa</i> 70 % . . . . .	93
Tabela 11 – Resultados para inserção do APA e EST com 90 % de margem de CC e <i>mapa</i> . . . . .	94
Tabela 12 – Resultados para inserção do APA e EST com 75 % . . . . .	109
Tabela 13 – Resultado de inserção para margem de 80 % . . . . .	109
Tabela 14 – Resultado de inserção para margem de 85 % . . . . .	110
Tabela 15 – Resultado de inserção para margem de 90 % . . . . .	110
Tabela 16 – Resultado de inserção para margem de 75 % e <i>mapa</i> 80 % . . . . .	110
Tabela 17 – Resultado de inserção para margem de 80 % . . . . .	110
Tabela 18 – Resultado de inserção para margem de 85 % . . . . .	111
Tabela 19 – Resultado de inserção para margem de 90 % . . . . .	111
Tabela 20 – Resultados para inserção do APA e EST com 75 % de margem de caminhos críticos e <i>mapa</i> 90 % . . . . .	111
Tabela 21 – Resultados para inserção do APA e EST com 80 % de margem CC e <i>mapa</i> 90 % . . . . .	112
Tabela 22 – Resultados para inserção do APA e EST com 85 % de margem CC e <i>mapa</i> 90 % . . . . .	112
Tabela 23 – Resultados para inserção do APA e EST com 90 % de margem de caminhos críticos e <i>mapa</i> . . . . .	112



# Lista de algoritmos

1	Inserção de APA e EST . . . . .	73
2	Frequência máxima do circuito . . . . .	81



# Lista de abreviaturas e siglas

APA	Ativação de Caminho Alternativo
APAFF	APA Flip-Flop
AVT	Alta tensão de limiar
BCD	<i>Binary-coded Decimal</i>
BTI	<i>Bias Temperature Instability</i>
BVT	Baixa tensão de limiar
CC	Caminho Crítico
CHE	<i>Channel Hot-Electron</i>
CI	Circuito Integrado
CK	<i>Clock</i>
CMOS	<i>Complementary Metal-Oxide-Semiconductor</i>
CNC	Caminho Não-Crítico
DAHC	<i>Drain Avalanche Hot-Carrier</i>
DSP	<i>Digital Signal Processor</i>
DT	Detector de Transição
EDA	<i>Electronic Design Automation</i>
EDTF	Escalonamento Dinâmico de Tensão e Frequência
EST	Empréstimo Seletivo Temporal
FET	Transistor de Efeito de Campo
FF	Flip-Flop
GALS	Globalmente-Assíncrono Localmente-Síncrono
GHz	Gigahertz
HBD	<i>Hard Breakdown</i>

HCI	<i>Hot Carrier Injection</i>
HDL	<i>Hardware Description Language</i>
MD	<i>Metastability Detector</i>
MHz	Megahertz
MMSIM	<i>Multi-Mode Simulation with Spectre</i>
MUX	Multiplexador
NBTI	<i>Negative Bias Temperature Instability</i>
nMOS	Transistor FET do tipo <i>Metal-Oxide-Semiconductor</i> com canal N
PCB	<i>Printed Circuit Board</i>
pMOS	Transistor FET do tipo <i>Metal-Oxide-Semiconductor</i> com canal P
RAM	<i>Random Access Memory</i>
RMN	Redundância Modular N
RMT	Redundância Modular Tripla
RTL	<i>Register Transfer Logic</i>
SGHE	<i>Secondary Generated Hot-Electron</i>
SHE	<i>Substrate Hot-Electron</i>
SRAM	<i>Static Random Access Memory</i>
STA	<i>Static Timing Analysis</i>
TBFF	<i>Time-Borrowing</i> Flip-Flop
Tcl	<i>Tool Command Language</i>
TDDB	<i>Time Dependent Dielectric Breakdown</i>
TG	<i>Transmission Gate</i>
ULA	Unidade Lógica Aritmética
VLSI	<i>Very Large Scale Integration</i>

# Sumário

<b>1</b>	<b>Introdução</b>	<b>21</b>
1.1	Contextualização	22
1.2	Objetivos	24
1.3	Escopo	25
1.4	Organização do Trabalho	25
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>27</b>
2.1	A Física por Trás das Falhas	28
2.1.1	Hot Carrier Injection	28
2.1.2	Time-Dependent Dielectric Breakdown	29
2.1.3	Eletromigração	29
2.1.4	Bias Temperature Instability	30
2.2	Tipos de Falhas	31
2.2.1	Falhas Transientes	32
2.2.2	Falhas Permanentes	32
2.2.3	Falhas Intermitentes	33
2.3	Técnicas Básicas Contra Falhas	33
2.3.1	Redundâncias Espaciais	33
2.3.2	Redundâncias Temporais	37
2.3.3	Redundâncias de Informação	37
2.4	Circuitos Sequenciais	38
2.4.1	Elementos Sequenciais Ideais	38
2.4.2	Elementos Sequenciais Reais	40
2.4.3	Falha de Temporização	44
2.5	Fluxo de Projeto de Circuitos Digitais	45
2.6	Estado da Arte	48
2.6.1	Mascaramento de Erros Com Elementos Sequenciais	50
<b>3</b>	<b>Metodologia</b>	<b>57</b>
3.1	Visão Geral e Motivação	57
3.2	Detecção de Erros	59
3.3	Empréstimo Seletivo Temporal	61
3.4	Ativação de Caminho Alternativo	64
3.4.1	Estrutura do APAFF	70
3.5	Fluxo de Inserção	72
<b>4</b>	<b>Simulações e Resultados</b>	<b>75</b>
4.1	Validação das Técnicas	75
4.1.1	Circuitos de Validação	77

4.1.2	Metodologia de Simulação . . . . .	80
4.1.3	Resultados de Simulação . . . . .	80
4.1.4	Metaestabilidade na Validação . . . . .	87
4.2	Inserção em Circuitos Complexos . . . . .	90
4.2.1	Métricas de Inserção . . . . .	90
4.2.2	Resultados de Inserção . . . . .	91
4.3	Processador ARM . . . . .	97
<b>5</b>	<b>Conclusão . . . . .</b>	<b>101</b>
5.1	Objetivos Alcançados . . . . .	103
5.2	Trabalhos Futuros . . . . .	103
	<b>Referências . . . . .</b>	<b>105</b>
	<b>APÊNDICE A Tabelas de Medições . . . . .</b>	<b>109</b>

# 1 Introdução

Avanços na microeletrônica e microfabricação levaram os Circuitos Integrados (CIs) a altos níveis de complexidade e desempenho em aplicações como computação e sistemas eletrônicos embarcados. A força motriz por trás desses avanços é a redução dos dispositivos básicos usados na construção dos CIs, os transistores, que já alcançam escalas nanométricas.

A densidade de transistores em um único chip vem aumentando em cerca de 35 % ao ano, duplicando, em média, a cada 2 anos. O aumento na área do circuito (*die size*) chega a 20 % ao ano, combinando em um aumento em cerca de 40 % a 50 % no número absoluto de transistores em um único chip nesse período de tempo. Essa tendência é conhecida popularmente por lei de Moore e a indústria de semicondutores vem avançando cadenciada por essa progressão (HENNESSY; PATTERSON, 2011).

Tabela 1 – Avanço tecnológico em processadores

Processador:	Intel 80286	Intel Pentium	Intel Pentium 4	Intel Core i7
Ano:	1982	1993	2001	2010
Área (mm <sup>2</sup> ):	47	90	217	240
Transistores:	134.000	3.100.000	42.000.000	1.170.000.000
Clock (MHz):	12,5	66	1500	3333
Tecnologia:	1,5 $\mu$ m	0,8 $\mu$ m	90nm	32nm

Adaptado de (HENNESSY; PATTERSON, 2011)

A Tabela 1 mostra o avanço em processadores. A redução nas dimensões dos transistores possibilitou maior agregação de funções e melhor desempenho desses chips. Outros dispositivos integrados, como memórias RAM e Flash, testemunharam igual progresso na integração em larga escala, possibilitando maior capacidade de armazenamento e menores tempos de acesso.

As conseqüências dessa miniaturização, no entanto, vem se tornando cada vez mais aparentes. Variações nos parâmetros dos dispositivos microeletrônicos podem ser induzidas tanto por variações nos processos de fabricação, como por desgaste durante a vida operacional do circuito. Esses efeitos comprometem o comportamento dos CIs, desviando seus parâmetros de operação dos dimensionados pelos projetistas e afetando as aplicações onde são empregados. Soluções para mitigar os efeitos dessas variações podem ser aplicadas em nível estrutural, a partir de novas tecnologias e dispositivos integrados, ou em nível de projeto com a aplicação de técnicas de confiabilidade (BORKAR, 2005). O progresso das tecnologias de circuitos integrados tende a acompanhar a lei de Moore

através de novas implementações, como circuitos em GaN, SiC e optoeletrônica. Essas novas tecnologias, no entanto, não possuem o amadurecimento necessário para serem empregados em larga escala, principalmente em aplicações que exigem alta confiabilidade ([BENSOUSSAN, 2017](#)).

O presente trabalho apresenta duas técnicas de confiabilidade desenvolvidas com o propósito de aumentar a vida útil de circuitos integrados digitais. A primeira realiza realocação de folgas temporais para aumentar a margem de tempo de processamento em caminhos lógicos críticos, intitulada Empréstimo Seletivo Temporal (EST). A segunda é empregada em situações onde não existe essa folga, utilizando uma redundância espacial maior para restaurar a máquina em tempo de execução, chamada Ativação de Caminho Alternativo (APA, do inglês *Alternative Path Activation*). Ambas as técnicas não requerem suspensão da máquina e são de uso geral, podendo ser empregadas não somente em processadores, mas também em máquinas que contenham dois ou mais estágios sequenciais.

A seção a seguir apresenta uma breve contextualização dos problemas enfrentados por circuitos digitais, principalmente computadores, na realização de processamentos confiáveis na atualidade. Em seguida, uma seção apresentará os objetivos principais desse trabalho. O escopo será abordado em seguida, onde é explicitado o que será discutido ao longo deste trabalho. Por fim, a última seção irá expor a organização deste documento para facilitar o entendimento.

## 1.1 Contextualização

A invenção de componentes em estado sólido melhorou a confiabilidade de sistemas de computação digital consideravelmente. Computadores utilizando válvulas e relés já empregavam técnicas de tolerância a falhas devido a natureza de suas aplicações. Sistemas de comutação de telefonia, aplicações militares, monitoramento comercial e sistemas de controle exigiam uma computação confiável para garantir serviços e obter precisão ([SERLIN, 1984](#)). O primeiro computador comercial, UNIVAC I, já utilizava verificação de paridade de bits e contava com duas Unidades Lógicas Aritméticas (ULAs) para comparação de resultados. Hoje em dia, o aumento da dependência em sistemas de computação tornou-se tão grande que é impossível retornar para mecanismos menos sofisticados ([SIEWIOREK; SWARZ, 1992](#)).

O aumento da presença de circuitos microeletrônicos em aplicações críticas gera uma grande demanda por confiabilidade desses dispositivos. As aeronaves Boeing 777 e o Airbus 330, por exemplo, utilizam redundâncias triplas em seus computadores internos. Esses sistemas são críticos na proteção dos passageiros a bordo e exemplificam a demanda por sistemas confiáveis ([SHOUMAN, 2001](#)).

A ubiquidade de computadores, e circuitos integrados no geral, trouxe quatro

fatores que aumentam a necessidade por confiabilidade (SIEWIOREK; SWARZ, 1992):

- **Ambiente hostis:** Aplicações remotas, como telefonia, usufruem de sistemas computacionais localizados em regiões com baixa disponibilidade de manutenção. Condições ambientais, como altas temperaturas de operação, umidade e flutuações na alimentação, comprometem ainda mais a resistência desses sistemas.
- **Aumento dos custos de manutenção:** A queda nos preços de componentes eletrônicos e microeletrônicos foi acompanhada pelo aumento do custo da mão de obra especializada que presta serviços de manutenção para esses equipamentos. Usuários de sistemas complexos optam sempre por dispositivos que requerem baixa manutenção para mitigação desses custos.
- **Usuários inexperientes:** A medida que computadores se tornam mais universais, o nível de conhecimento dos usuários que os operam diminui. Os sistemas de computação devem ser mais tolerantes a falhas oriundas de sua má operação.
- **Sistemas mais complexos:** As chances de um sistema falhar são diretamente proporcionais à probabilidade de falha de suas partes individuais. A medida que transistores se tornam menores e possibilitam maior integração de funções em um único chip, aumenta-se a probabilidade de falhas em um CI.

Além desses fatores, a crescente miniaturização de componentes resulta em maior susceptibilidade destes às variações no processo de fabricação. Os processos litográficos utilizados na produção de circuitos integrados possuem variações inerentes que são transmitidas para os *wafers* (pastilhas de silício onde os chips são fabricados). Essas são divididas em variações *inter-die* e *intra-die*. As primeiras afetam diferentes circuitos dentro de um único *wafer* e resultam em variações na velocidade de operação para circuitos de um mesmo tipo. Por afetarem *dies* inteiros, variações *inter-die* são, geralmente, menos problemáticas. (BERNSTEIN et al., 1998).

Variações *intra-die* ocorrem dentro de um mesmo circuito, na forma de variações dos parâmetros elétricos dos componentes e interconexões (YE et al., 2011). Incompatibilidades temporais na lógica podem surgir por causa dessas variações. Com isso em mente fabricantes fornecem diferentes modelos de transistores contendo os parâmetros do melhor e pior caso esperado de seus processos. Dessa forma, o projeto do circuito pode ser feito levando-se em conta variações na fabricação dos chips. A abordagem usual no projeto é a inserção de margens de tensão e frequência para garantir que o circuito funcionará dentro das variações inseridas pelos processos (BERNSTEIN et al., 1998).

Além da fabricação, efeitos de degradação se tornaram um problema para CIs. A redução das dimensões dos transistores, principalmente da camada dielétrica dos FETs

(*Field Effect Transistor*), resultou em um aumento dos campos elétricos dentro dos componentes. Efeitos como *bias temperature instability*, injeção de portadoras energizadas e rompimento do dielétrico passaram a afetar o funcionamento de circuitos integrados e a reduzir sua vida útil (HUARD; DENAIS; PARTHASARATHY, 2006; TAKEDA; SUZUKI; HAGIWARA, 1983; STATHIS, 2001).

Para enfrentar esses efeitos e aumentar a vida útil, a indústria usualmente emprega abordagens pessimistas, como margens de frequência/tensão e redundâncias espaciais/temporais. Estas abordagens resultam em aumento significativo de área e consumo dos dispositivos microeletrônicos, maior custo de operação e menor desempenho, influenciando na própria especificação dos dispositivos (SORIN, 2009).

Visando reduzir essas margens pessimistas, soluções são propostas envolvendo dispositivos especiais que detectam e corrigem falhas localmente nos circuitos. Técnicas como o RAZOR (ERNST et al., 2003), RAZORII (DAS et al., 2009) e TIMBER (CHOUDHURY et al., 2010) visam aumentar a robustez do sistema ao inserir elementos de memória com capacidade de detectar sinais atrasados e corrigi-los *in loco*. Essas e outras técnicas são apresentadas e discutidas neste trabalho.

Nas técnicas propostas o sistema não precisa ser suspenso ou restaurado para que haja recuperação do erro e ambas são compatíveis com circuitos de propósito específico, não somente processadores. As técnicas podem ser inseridas automaticamente no circuito, com o mínimo de interação do usuário graças a um fluxo automático também apresentado neste trabalho.

O EST e APA foram desenvolvidos e testados utilizando um conjunto especial de ferramentas para projeto e simulação de circuitos. O *software* Cadence Virtuoso foi empregado na validação das técnicas, verificando seu comportamento em nível elétrico. A inserção das técnicas em circuitos digitais de maior escala foi realizada através do *software* Cadence RTL Compiler (BRUNVAND, 2010). Nessa ferramenta, um programa foi escrito em linguagem Tcl que realiza a captação de caminhos críticos, inserção das técnicas no circuito e avaliação da cobertura e ganho alcançados.

## 1.2 Objetivos

Junto a contextualização estabelecida, os principais objetivos deste trabalho podem ser definidos como:

- Realizar uma revisão bibliográfica capaz de abarcar os principais conceitos e técnicas utilizadas pelo estado da arte;
- Elaborar duas técnicas capazes de aumentar a vida útil de circuitos integrados digitais;

- Analisar as técnicas por meio de simulações;
- Elaborar um fluxo de inserção automático das técnicas;
- Quantificar seus benefícios em circuitos de teste.

## 1.3 Escopo

A área de confiabilidade em CIs é vasta. Muitas técnicas e abordagens são utilizadas em diferentes aplicações, topologias e tipos de circuitos. Para facilitar o entendimento, esta seção descreve o escopo do trabalho desenvolvido, ajudando a determinar o que será encontrado nas próximas páginas deste texto.

As técnicas propostas, assim como a pesquisa realizada na literatura, focam em circuitos integrados digitais. Isso é, chips construídos com componentes micrométricos em integração de muito grande escala (VLSI). Assim, o desenvolvimento deste trabalho foi direcionado a microprocessadores e outros dispositivos de propósito geral, apesar das técnicas em si não serem limitadas somente a estes.

Os tópicos são abordados com ênfase no nível lógico dos circuitos. Apesar de ocasionalmente serem necessárias as explicações em nível físico (transistores, capacitores, entre outros), estas foram mantidas curtas e de forma simplificada. Na Seção 2.1, por exemplo, os efeitos de degradação são explicados qualitativamente, possibilitando o entendimento dos mecanismos físicos sem desviar o foco das manifestações lógicas desses problemas.

A revisão bibliográfica conta com técnicas encontradas comercialmente e outras experimentais desenvolvidas para lidar com problemas de confiabilidade. O foco principal do tópico de estado da arte é em técnicas utilizando elementos sequenciais (flip-flops e latches) especiais, capazes de detectar e corrigir erros temporais. Esse foco é necessário devido a natureza das técnicas propostas, que se encaixam nessa categoria.

Algoritmos foram utilizados para explicar os fluxos automáticos desenvolvidos ao longo do trabalho de forma a facilitar o entendimento das rotinas que, na prática, foram implementadas com linguagens de programação específicas (como Tcl e SKILL). Os resultados obtidos pelas simulações são fornecidos de forma a quantificar os benefícios e custos de implementação das técnicas. Uma discussão acerca dos valores é fornecida para facilitar o entendimento dos resultados apresentados.

## 1.4 Organização do Trabalho

Este texto organiza-se da seguinte forma:

- **Capítulo 2 – Revisão Bibliográfica:** Fundamentação teórica necessária para o entendimento das técnicas propostas por este trabalho. São discutidos efeitos de degradação, técnicas de confiabilidade empregadas usualmente na indústria e estado da arte;
- **Capítulo 3 – Metodologia:** Uma visão geral é dada sobre as técnicas encontradas no estado da arte. O Empréstimo Seletivo Temporal e a Ativação de Caminhos Alternativos são apresentados e explicados nessa seção, junto com a forma como são inseridas em circuitos complexos.
- **Capítulo 4 – Simulações e Resultados:** Neste capítulo são apresentados as metodologias empregadas para simular as técnicas, circuitos de teste e os resultados obtidos. Uma análise da aplicabilidade das técnicas também é feita.
- **Capítulo 5 – Conclusão:** O capítulo de conclusão encerra este trabalho, resumindo os tópicos abordados e os resultados obtidos junto a uma discussão de viabilidade.

## 2 Revisão Bibliográfica

Durante muitos anos, o foco do desenvolvimento de circuitos digitais - principalmente computadores - foi o desempenho, liderado pela disponibilidade de transistores cada vez mais rápidos e menores. No entanto, a lei de Moore tem enfrentado diversos desafios para se manter verdadeira devido à miniaturização mais agressiva dos componentes microeletrônicos. Com isso, a área de sistemas confiáveis tem se tornado mais importante para o desenvolvimento de dispositivos que cumpram os seus objetivos de forma segura (SORIN, 2009).

Um sistema tolerante a falhas é aquele capaz de continuar operando mesmo na presença de problemas de hardware ou de software. Sistemas assim são requisitados em diversas situações, como em aplicações de vida longa, computações críticas, sistemas em condição de pouca manutenção e aplicações de alta disponibilidade.

As aplicações de vida longa se caracterizam pela necessidade de autonomia dos componentes envolvidos. Um bom exemplo desse tipo de aplicação são satélites e outras naves não tripuladas. Um requerimento típico desses dispositivos é o de possuir 95% de chance de ainda estar operando após 10 anos de serviço (JOHNSON, 1996).

Computações críticas são, talvez, as aplicações mais conhecidas de computação tolerante a falhas. Esses são os sistemas onde a operação correta é essencial para a segurança da vida humana, prevenção de poluição e de desastres com grande perda material. Sistemas de controle de tráfego aéreo e alguns controladores industriais são exemplos de computações críticas que demandam tolerância a falhas.

Baixa disponibilidade de manutenção é uma situação comum para dispositivos que operam em regiões ermas e longe de qualquer tipo de suporte físico e reposição de peças. Estações de telefonia ou subestações de transmissão de energia são exemplos de sítios com dispositivos que podem não receber manutenção por longos períodos de tempo. Computação tolerante a falhas se torna essencial para esses sistemas manterem sua operação normal até que a manutenção possa ser providenciada.

Alta disponibilidade é um parâmetro chave em muitas aplicações, como sistemas bancários. Usuários desses serviços demandam alta probabilidade de terem suas requisições atendidas, não deixando margem a falhas do sistema (JOHNSON, 1996).

Essa seção está dividida em vários tópicos que abordam diferentes aspectos dos problemas que afligem aos sistemas digitais usualmente empregados. Na Seção 2.1 uma introdução é dada a cerca dos fenômenos físicos que levam às falhas e mudanças de comportamento dos dispositivos microeletrônicos. Em seguida, na Seção 2.2, os tipos de

falhas e suas manifestações são apresentadas. As técnicas básicas usualmente empregadas na indústria são percorridas na Seção 2.3. Uma revisão sobre circuitos sequenciais e seus principais parâmetros utilizados no dimensionamento de sistemas digitais modernos é feita na Seção 2.4. Por último, um tópico de estado da arte apresenta os principais trabalhos voltados à confiabilidade desses sistemas.

## 2.1 A Física por Trás das Falhas

Com a redução dos transistores à escala nanométrica, a confiabilidade dos circuitos microeletrônicos se tornou o assunto dominante no que diz respeito à vida útil e operação desses dispositivos (FIROUZI et al., 2015).

Os principais efeitos relacionados à degradação de circuitos integrados são estudados desde os anos setenta e oitenta. Na época, as pesquisas eram voltadas para o entendimento desses efeitos ao invés de prover soluções para circuitos, uma vez que esses efeitos não eram significativos. Com a redução da geometria dos transistores e o conseqüente aumento de campo elétrico no dielétrico entre porta e canal, o assunto de degradação passou a se tornar uma preocupação maior.

A degradação física de um circuito pode ocorrer por diversos fatores. Entre eles o desgaste de fios através do efeito de eletromigração, desvio de tensão limiar por *Bias Temperature Instability* (BTI), rompimento do dielétrico de porta do transistor com o tempo (*Time Dependant Dielectric Breakdown*, ou TDDB) e injeção de portadores com alta energia cinética na interface entre dielétrico e silício (*Hot Carrier Injection*, HCI) são tidos como os principais. Estes efeitos são especialmente acentuados em tecnologias abaixo de 90nm (MARICAU; GIELEN, 2013).

### 2.1.1 Hot Carrier Injection

O efeito de HCI é definido por portadores com alta energia cinética, ou *hot carriers*. Esses portadores adquirem energia a partir de alto campo elétrico no canal do transistor presente em duas orientações: Um vertical porta-canal e um horizontal dreno-fonte. Portadores com energia alta o suficiente podem ser injetados em regiões proibidas da interface entre o dielétrico e o silício, ao invés de percorrerem seu trajeto entre fonte e dreno. Quando essa injeção ocorre surgem estados na interface, que geram desvios nas características elétricas do transistor como tensão de limiar  $V_{th}$ , fator de corrente  $\beta$  e condutância de saída  $g_0$ . O HCI é um efeito mais comum a transistores do tipo nMOS (MARICAU; GIELEN, 2013).

Os mecanismos de injeção de hot carriers na interface do óxido são divididos em quatro (TAKEDA; SUZUKI; HAGIWARA, 1983):

- Injeção por elétron de canal (*channel hot-electron*, CHE) é composto por elétrons chamados de “sortudos”. Quando a tensão de dreno  $V_d$  é próxima da tensão de porta, elétrons podem ganhar energia suficiente para romper a barreira da interface do óxido e ficarem presos;
- Injeção por elétrons de substrato (*substrate hot-electron*, SHE) ocorre por elétrons que adentram o canal vindo do corpo (*bulk*) do transistor, atraídos pelo campo elétrico vertical da porta;
- Injeção por avalanche de portadores no dreno (*drain avalanche hot-carrier*, DAHC) ocorre com o efeito de avalanche entre portadores na região do dreno, afetando tanto buracos quanto elétrons;
- Injeção por elétrons gerados secundariamente (*secondary generated hot electron injection*, SGHE) envolve a produção de portadores com alta energia cinética a partir de impactos ionizantes entre portadores secundários. Esse efeito possui uma menor influência na degradação dos transistores.

### 2.1.2 Time-Dependent Dielectric Breakdown

A fina camada de material dielétrico situada entre a porta e o canal do transistor possibilita o controle elétrico do dispositivo. A fina espessura que essa camada possui em tecnologias modernas a vulnerabiliza diante de campo elétrico imposto pela porta. Quando um campo elétrico alto o suficiente é aplicado à porta, o rompimento do dielétrico pode ocorrer de maneira violenta e repentina, chamado *hard breakdown* (HBD). O HBD resulta em perda das propriedades isolantes de óxido e aumento repentino da corrente de porta (MARICAU; GIELEN, 2013).

A degradação progressiva do dielétrico com o tempo (*Time-Dependent Dielectric Breakdown*, TDDDB) é induzida por campos elétricos menores. Quando esse campo é aplicado sobre uma camada dielétrica fina o suficiente, uma corrente fluirá entre porta e canal devido ao efeito de tunelamento quântico. Essa corrente acionará diversos efeitos ao longo da barreira dielétrica. O mais significativo entre eles, para baixas tensões de porta, é a criação de armadilhas ionizadas através da liberação de hidrogênio no dióxido. Os átomos de hidrogênio livres dentro do dielétrico levam a redução de suas propriedades de isolante. Quando esse dano é grande o suficiente, as propriedades dielétricas do meio começam a falhar e a permitir a condução entre porta e silício (STATHIS, 2001).

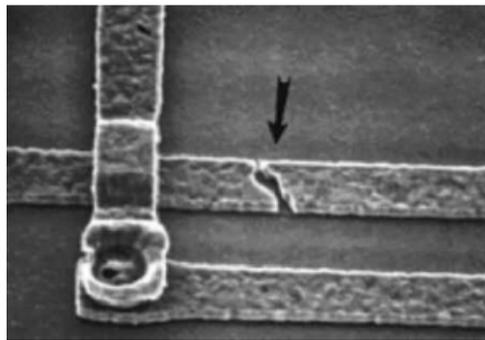
### 2.1.3 Eletromigração

O modelo do elétron livre para condução em metais assume que os elétrons se movem ao longo do material sem restrições devido a uma estrutura metálica perfeita. Essa

aproximação só é precisa quando se assume uma relativa baixa densidade de corrente no condutor. Um fio comum, como o de uma extensão elétrica doméstica por exemplo, possui uma densidade usual de corrente em torno de quatro ordens de grandeza inferior à uma trilha metálica em um chip. A dispersão de elétrons no condutor resulta, sob essa densidade, no deslocamento de átomos do metal na mesma direção que o fluxo eletrônico. A esse deslocamento de material devido a corrente elétrica é chamado de eletromigração (TU, 2003).

O deslocamento de material causado pela eletromigração gera buracos no material condutor na região onde os átomos foram removidos e acúmulo de material onde são depositados. Essa deformação das trilhas resulta no aumento de resistência das interconexões e na acentuação do próprio efeito de eletromigração, podendo acarretar no rompimento completo da conexão elétrica, como pode ser visto na Figura 2.1.

Figura 2.1 – Efeito da eletromigração em uma trilha metálica



Fonte: (OVERCLOCKING.GUIDE, 2014)

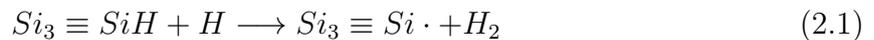
Os modelos utilizados industrialmente para descrever os efeitos da eletromigração contavam somente com temperatura e densidade de corrente, chamado modelo de *Black* (BLACK, 1967). Contudo, experimentos atuais mostram que o modelo de Black não é preciso o suficiente para os níveis de miniaturização alcançados atualmente e novos modelos para a sua substituição já são propostos (JING; LIANG; MENG, 2010).

#### 2.1.4 Bias Temperature Instability

Este efeito tem recebido bastante atenção devido ao seu impacto mais acentuado em tecnologias nanométricas. O BTI manifesta-se principalmente como um desvio gradativo da tensão de limiar dos transistores FETs quando em altas temperaturas e sob campos elétricos relativamente pequenos. Essas características fazem com que o BTI se manifeste em um circuito antes do HCI, sendo necessária somente uma elevada temperatura de operação (entre  $100^{\circ}\text{C}$  e  $250^{\circ}\text{C}$ ), característica bastante comum em SoC (*System on Chip*) e circuitos digitais modernos (SCHRODER; BABCOCK, 2003).

O efeito causado pelo BTI manifesta-se principalmente em FETs do tipo P, sendo chamado de NBTI (*Negative Bias Temperature Instability*). O NBTI é um dos principais responsáveis no aumento do atraso em lógicas combinacionais de circuitos digitais, levando a problemas de operação dos mesmos (FIROUZI et al., 2015).

Um átomo de silício conecta-se tetraedricamente com seus vizinhos no corpo do *wafers* onde os transistores são fabricados. Quando uma camada de dióxido de silício ( $SiO_2$ ) é criada para compor o dielétrico de porta, os átomos de Si da superfície passam a realizar ligações com oxigênio e hidrogênio. Se durante a operação o dispositivo encontra-se sob alta temperatura, átomos livres de hidrogênio podem se recombinar com hidrogênios conectados aos átomos de Si na interface, deixando para trás uma carga elétrica que compõe uma armadilha.



Na reação descrita em 2.1, o símbolo “ $\equiv$ ” representa três ligações covalentes entre  $Si_3$  e  $Si$  e “ $\cdot$ ” representa um elétron livre em órbita do silício da interface. O acúmulo dessas cargas na superfície resulta em um deslocamento da tensão de limiar para valores mais negativos em transistores do tipo pMOS, resultando em uma resposta mais lenta do dispositivo (SCHRODER; BABCOCK, 2003).

## 2.2 Tipos de Falhas

Problemas em circuitos se apresentam em três formas: Falhas, erros e falhas de sistema. A falha é um defeito físico ou imperfeição presente em um componente de hardware. O erro é a manifestação da falha em nível lógico, ou seja, é o desvio de acurácia ou a apresentação errônea de um resultado. Por fim, a falha de sistema ocorre quando o erro resulta em uma função realizada incorretamente e que é aparente para o usuário.

Esses conceitos podem ser melhor apresentados ao se dividir os sistemas em três universos. O primeiro universo é o físico, onde estão contidos os dispositivos semicondutores, monitores, fontes de alimentação, entre outros. A falha é então a alteração física de um dispositivo que habita esse universo. O segundo universo é o informacional, onde ocorrem os erros. Erros afetam unidades de informação, ou bits, usados para interpretação de resultados e outros cálculos. O último universo é o de usuário. Neste universo, os erros se manifestam na forma de falhas de sistema (JOHNSON, 1996).

Os problemas que afligem sistemas microeletrônicos podem se manifestar com diferentes durações e a partir de efeitos físicos distintos. As falhas podem ser divididas em transientes, permanentes ou intermitentes.

### 2.2.1 Falhas Transientes

A falha transiente é aquela que não persiste no sistema, ocorrendo em um instante de tempo curto e desaparecendo em seguida. É o caso de falhas acarretadas por radiações cósmicas. Se uma partícula carregada atinge o chip com energia suficiente (denotada energia crítica,  $Q_{crit}$ ) em uma região de junção  $p - n$ , ocasionará em uma mudança no estado de condução de um transistor. Essa mudança se manifestará no nível lógico de sistema através da troca de um bit. Se esse bit estiver armazenado em uma unidade de memória (como uma SRAM ou flip-flops), essa falta poderá se propagar para níveis mais altos e se tornar uma falha de sistema (SORIN, 2009).

Outra fonte de falhas transientes é a interferência eletromagnética que pode advir de fontes externas ou internas. Uma trilha interna do chip pode induzir uma vizinha por acoplamento e ocasionar uma mudança no valor de um bit. Esse fenômeno também é conhecido por “cross-talk”.

### 2.2.2 Falhas Permanentes

Falhas permanentes podem ocorrer por três motivos: Degradação física, defeitos de fabricação ou falhas de projeto. Os dois primeiros motivos dependem de fatores construtivos e de processos fabris do circuito, sendo necessários o emprego de técnicas de confiabilidade e qualidade de produção para blindar o sistema contra falhas. Já o último depende da adoção de técnicas e abordagens para validação de projeto, que contam com passos de verificação para evitar que falhas de projeto se tornem erros durante a operação do circuito (SORIN, 2009).

O processo de fabricação de CIs é propenso a muitas falhas e imperfeições devido à natureza delicada dos métodos empregados. Esses defeitos podem ser identificados através dos processos de teste pós-fabricação, importantes para evitar o envio de chips com falhas para o consumidor final. Na análise e elaboração de sistemas tolerantes a falhas, defeitos de fabricação podem ser considerados iguais aos ocasionados por degradação, com a diferença de que o tempo de ocorrência é no início da vida útil do circuito. Além disso, essas falhas possuem uma chance muito maior de ocorrerem simultaneamente do que as causadas por degradação física.

Transistores e fios menores são mais suscetíveis a variações de processos de fabricação. A fotolitografia é um processo que conta com imperfeições que são transmitidas naturalmente para o chip sendo fabricado. Em gerações anteriores de CMOS, uma variação de  $2nm$  em uma tecnologia de  $250nm$  possui muito menos efeito do que a mesma variação em um processo de  $22nm$  (SORIN, 2009).

Falhas de projeto (também chamadas de *bugs*) podem afetar até mesmo os chips fabricados em perfeito estado, resultando em falhas de sistema durante sua operação. Um

exemplo relativamente conhecido desse tipo de falha é o caso de processador *Pentium*, da Intel, lançado no mercado com um problema em sua operação de divisão (BLUM; WASSERMAN, 1996). Desenvolver métodos resistentes a falhas de projetos representa um desafio único na área de pesquisa em sistemas tolerantes a falhas (SORIN, 2009).

### 2.2.3 Falhas Intermitentes

Falhas intermitentes se manifestam em intervalos de tempo geralmente curtos e desaparecem em seguida, voltando a acontecer quando alguma condição se cumpre. Um exemplo desse tipo de falha é uma conexão fracamente estabelecida através de um fio. Durante variações de temperatura, o ponto de má conexão pode apresentar alta resistência à passagem de corrente, gerando uma falha física. Em outros momentos, o fio pode apresentar uma conexão elétrica ininterrupta. Esses problemas se tornam mais frequentes com a alta densidade de componentes micro e nanoeletrônicos e a alta temperatura de operação dos chips modernos (CONSTANTINESCU, 2003).

## 2.3 Técnicas Básicas Contra Falhas

A forma mais básica de verificar se um erro aconteceu em um determinado resultado é através de redundâncias. O circuito realiza seus cálculos e depois compara os resultados com outros processamentos que emitem os mesmos valores. Discrepâncias nessas comparações são marcadas como erros.

Redundância implica no acréscimo de recursos, sejam eles informação, tempo ou capacidade de processamento, além do que o sistema necessita para sua operação normal. Redundâncias podem assumir diversas formas, como redundância espacial, *software*, informação ou de tempo. Por questões de escopo deste trabalho, as redundâncias de *software* e informação não serão abordadas.

### 2.3.1 Redundâncias Espaciais

O acréscimo de recursos de processamento, como um bloco funcional ou um processador extra, é uma forma simples de redundância espacial. Todos os elementos idênticos realizam um determinado cálculo e seus resultados são comparados em busca de erros. Essas redundâncias podem ser divididas em passivas, ativas ou híbridas. Nas primeiras, o erro é mascarado sem que haja detecção, isolamento ou manutenção do módulo defeituoso. Redundâncias ativas não mascaram as falhas, mas as detectam ativamente, localizam e substituem o componente defeituoso por um sobressalente. Por último, redundâncias híbridas juntam as vantagens das duas anteriores. Utiliza mascaramento de falhas para impedir a suspensão do sistema enquanto detecta e substitui o módulo defeituoso (JOHNSON, 1984).

As redundâncias passivas utilizam técnicas de mascaramento de falhas para gerar tolerância que não requer intervenção do operador ou sistema. Em sua forma mais básica, elas não detectam as falhas, apenas evitam que essas se propaguem para o nível de informação do sistema. O mascaramento de falhas pode ocorrer de forma não intencional em um circuito e em diferentes níveis (SORIN, 2009):

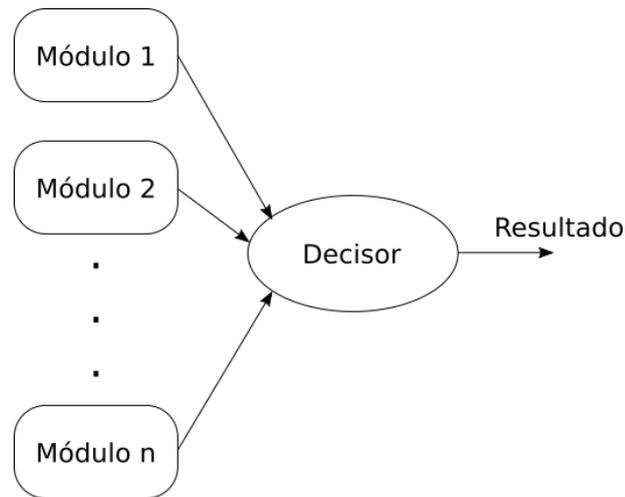
- Mascaramento lógico é quando um erro não é propagado em um circuito. Por exemplo, uma porta lógica AND que tenha um erro em uma entrada e um valor zero em outra, não propagará o erro adiante devido a sua tabela verdade;
- Mascaramento de arquitetura ocorre quando um erro não se manifesta na aplicação final de usuário. Um registrador que guarda um valor errado mas que nunca é lido, por exemplo, representa um mascaramento de arquitetura;
- Mascaramento de aplicação são falhas a nível de usuário que não afetam a operação de sistema. Um erro que ocorre em um espaço de memória utilizado pelo usuário mas que nunca é acessado pelo programa é um exemplo de mascaramento de aplicação. Essa falha nunca se manifestará enquanto não for acessada.

A forma mais básica de redundância passiva é a modular. Nela, três ou mais módulos idênticos - de preferência em número ímpar - têm suas entradas alimentadas com os mesmos valores. Um bloco decisor é conectado às saídas e este passa adiante o valor a partir de uma votação de maioria. Um exemplo dessa técnica é a Redundância Modular Tripla (RMT), onde três módulos iguais são utilizados. Na comparação de suas saídas, o resultado que possuir ao menos dois votos é considerado correto e, se houve um voto contra, o módulo que emitiu o resultado errado é marcado como defeituoso pelo sistema. De forma geral, quando mais de três módulos são utilizados, essa redundância recebe o nome de Redundância Modular N (RMN) (INSTRUMENTS, 2008). A Figura 2.2 mostra uma diagrama desse tipo de técnica.

O objetivo principal da redundância passiva é fornecer um resultado livre de erros e não de detectar o erro em si.

As redundâncias ativas - também chamadas dinâmicas - atuam através da detecção de falhas e da substituição dos dispositivos defeituosos por outros. São empregadas a detecção, localização e recuperação de falhas para realizar reconfigurações no sistema e aumentar a confiabilidade. Técnicas ativas, usualmente, não empregam mascaramento de falhas, sendo utilizadas em aplicações onde erros temporários e pausas para restauração do sistema são toleráveis para evitar os grandes acréscimos em área resultantes de redundâncias passivas. Como exemplo dessas aplicações pode-se citar sistemas digitais embarcados em satélites, que se beneficiam da menor área utilizada por redundâncias ativas (JOHNSON, 1988).

Figura 2.2 – Sistema RMN



Um sistema básico de redundância ativa é a duplicação com comparação. Nessa topologia, dois dispositivos realizam cálculos em paralelo e têm seus resultados comparados em busca de discrepâncias. Em sua forma básica, essa técnica não localiza erro, mas o detecta.

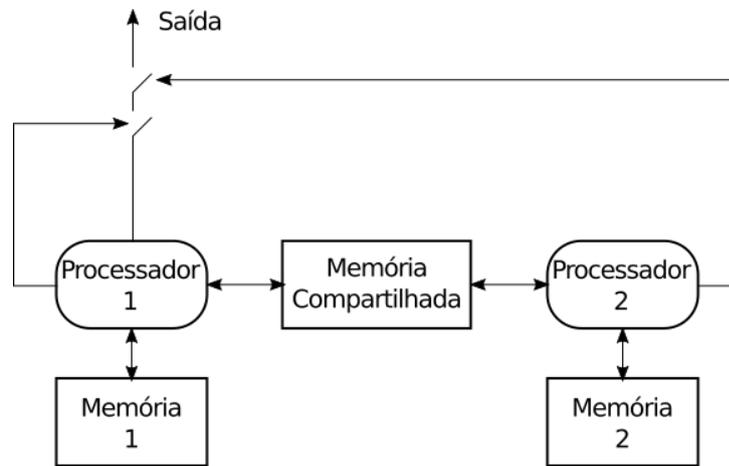
A comparação dos resultados de cada sistema podem ser executadas de diversas formas. A comparação bit-a-bit, empregada em sistemas simples, pode não ser a melhor forma de detectar erros. Em controle digital, onde sensores analógicos estão envolvidos, por exemplo, a comparação bit-a-bit pode sempre resultar em erro devido às diferenças entre sensores, temperaturas e arredondamentos.

A Figura 2.3 mostra um diagrama de dois processadores duplicados que comparam seus resultados em uma memória compartilhada. Essa comparação possibilita encontrar discrepâncias nos resultados emitidos. Se uma discrepância é detectada, a saída é interrompida e o sistema é notificado da falha.

Redundâncias ativas são usualmente acompanhadas de substituição do módulo com falha por um sobressalente. Quando um erro é detectado, um segundo dispositivo em repouso (*standby*) é inserido no sistema para substituir o defeituoso. Essa substituição pode ser feita através da inicialização de um sobressalente inativo, chamado de inicialização a frio (*cold standby*), ou de um sobressalente já em execução e pronto para assumir os processos, chamado de inicialização quente (*hot standby*). O primeiro possui a vantagem de poupar o dispositivo sobressalente de desgaste e garantir o seu bom funcionamento quando convocado, mas a sua inicialização pode ser demorada. Já o segundo possui maior desgaste por estar sempre em execução junto ao dispositivo principal, mas é mais rápido de inserir e dar continuidade aos processos em andamento (SORIN, 2009).

Redundâncias híbridas empregam métodos passivos e ativos de forma a aproveitar as vantagens de ambas. As técnicas híbridas utilizam mascaramento de falhas para evitar

Figura 2.3 – Redundância ativa dupla entre dois processadores com memória compartilhada

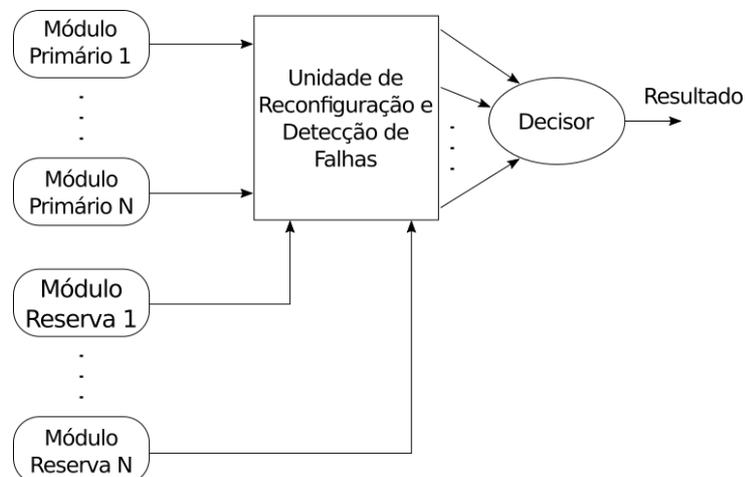


Fonte: Adaptado de Johnson (1988)

que resultados errados se propaguem. Detecção, localização e recuperação de falhas são utilizados para a substituição de dispositivos defeituosos por sobressalentes. Técnicas híbridas são usualmente empregadas em sistemas mais críticos devido aos grandes custos envolvidos (JOHNSON, 1996).

Uma forma de se implementar redundância híbrida é através de RMN com sobressalentes para substituição de módulos defeituosos. Um sistema com RMT e um módulo sobressalente, por exemplo, poderá mascarar o primeiro erro gerado através de votação, substituir o módulo defeituoso pelo sobressalente e ainda ser capaz de mascarar um próximo erro. O diagrama da Figura 2.4 mostra um sistema híbrido tipo RMN com N módulos sobressalentes para substituição quando uma falha é detectada.

Figura 2.4 – Diagrama de redundância híbrida com substituição de módulos defeituosos por módulos reserva



Fonte: Adaptado de Johnson (1988)

### 2.3.2 Redundâncias Temporais

Redundâncias espaciais requerem um aumento considerável de área e consumo. Os métodos de redundância temporal tentam remediar essa questão utilizando tempo como recurso para detecção de falhas. Esse tipo de redundância envolve a execução de um único processo múltiplas vezes na tentativa de detectar discrepâncias entre os resultados.

Em sua forma básica, uma redundância temporal é capaz de identificar falhas transientes. Um erro que afeta determinado valor devido a uma partícula com alta energia dificilmente se manifestará novamente, de forma idêntica, quando um reprocessamento for realizado. Falhas permanentes, no entanto, farão com que o dispositivo sempre chegue aos mesmos resultados errôneos. A detecção dessas falhas exigem mais recursos do que o simples reprocessamento (SORIN, 2009).

### 2.3.3 Redundâncias de Informação

A inserção de informação redundante em dados possibilita a detecção e mascaramento de erros em um sistema. Códigos específicos são empregados para reconstruir palavras e verificar sua validade, podendo detectar se houve um erro em sua manipulação ou não. Diferentes códigos existem para detectar e corrigir erros, variando em capacidade e complexidade.

Uma forma básica de detecção de erros ocorre ao se invalidar algumas combinações de bits. Um número representado em código BCD com 4 bits, por exemplo, possui somente dez combinações válidas que vão de 0 à 9 (0000 à 1001 em representação binária). Nesse caso um erro é detectado se, após a manipulação da informação, o número resultante aparecer fora dessa faixa. Essa forma de verificação é de simples implementação, bastando apenas um circuito lógico de verificação.

No entanto, o sistema com verificação por código BCD não é capaz de detectar o erro se uma mudança de bit mantém o número dentro da faixa de valores permitidos, sendo assim uma solução pouco robusta (JOHNSON, 1988).

Um código de correção de erros, por outro lado, possui a capacidade de detectar e restaurar o dado para o seu estado correto. Esses códigos são separados, usualmente, pela quantidade de bits que conseguem corrigir. Um código capaz de recuperar um dado de um bit errado é um código de correção única; se o dado pode ser restaurado de até dois bits errados, o código utilizado é de correção dupla e assim por diante. Esses códigos requerem maior processamento e impactam no custo do sistema, seja na forma de maior área/consumo ou em tempo de processamento.

Um conceito muito utilizado em códigos de detecção e correção é a distância de Hamming, que define quantos bits diferem entre duas palavras. O valor 0011 está a uma distância de Hamming 2 de 0000, pois dois bits são diferentes.

A distância de Hamming é muito útil na detecção e correção de erros através da redundância de informação. Considere um valor de 8 bits que só pode adotar quatro valores: 00000000, 00001111, 11110000 e 11111111. Se após uma manipulação ou transmissão de informação o dado recebido é 00000111, não só é possível detectar que um erro ocorreu, como também recuperar o valor original 00001111. Nesse caso, o erro ocorreu em um único bit e o valor correto foi restaurado a partir da menor distância de Hamming entre um valor válido e o recebido (no exemplo, distância 1) (TANENBAUM, 2011).

De um modo geral, um código capaz de corrigir  $c$  bits errados e detectar até  $d$  erros segue a Equação 2.2.

$$2c + d + 1 \leq H_d \quad (2.2)$$

O valor de  $H_d$  é a distância de Hamming entre dois códigos válidos (JOHNSON, 1988).

## 2.4 Circuitos Sequenciais

Além das falhas que podem ocorrer em circuitos integrados e as técnicas comumente empregadas na indústria para tratá-las, é importante se analisar o arranjo e organização interno desses circuitos. Nesta seção serão discutidos circuitos sequenciais temporizados por um sinal de sincronismo (chamado *clock*). Estes circuitos são a topologia básica para a construção de sistemas complexos, como processadores, DSPs (*Digital Signal Processor*) e outros sistemas digitais síncronos.

A lógica na utilização de circuitos sequenciais vem da necessidade de temporização dos dados e resultados. Para que uma dada computação seja executada, é necessário que os seus vários elementos estejam prontos um após o outro para se estabelecer uma ordem de execução.

Circuitos sequenciais são definidos assim pela presença de elementos de memória (às vezes chamados de elementos sequenciais) armazenando estados processados pela lógica combinacional. Os dois principais são o latch e flip-flop.

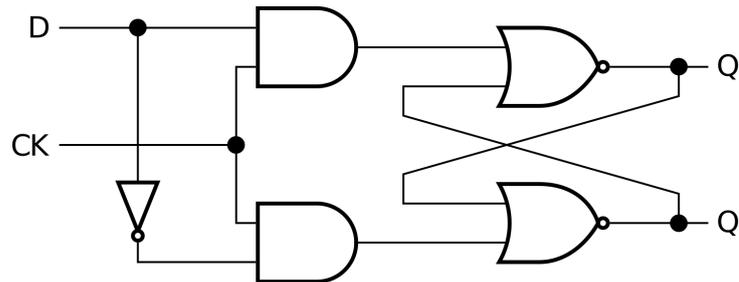
### 2.4.1 Elementos Sequenciais Ideais

Elementos sequenciais podem vir em diferentes topologias de implementação e entradas, como flip-flops JK, *set-reset*, tipos T e D. A título de simplificação, os elementos abordados neste trabalho serão sempre do tipo D. O latch tipo D é composto pelas entradas de dado  $D$  e de sincronismo, ou *clock*,  $CK$  e uma saída  $Q$  de forma que, idealmente,  $Q = D$  quando  $CK = 1$  (chamado de estado transparente). Quando  $CK = 0$ ,  $Q$  possuirá o valor que  $D$  tinha quando ocorreu a transição em  $CK$  de 1 para 0.

Tabela 2 – Tabela verdade do latch

D	CK	Q
0	0	Q
0	1	0
1	0	Q
1	1	1

Figura 2.5 – Diagrama do latch tipo D



A Tabela 2 mostra o comportamento de um latch padrão. O valor “Q” na coluna de saída representa a retenção do último valor que “D” tinha quando o sinal “CK” foi de “1” para “0”. O diagrama na Figura 2.5 mostra a estrutura lógica do latch tipo D.

O flip-flop tipo D possui as entradas  $D$ ,  $CK$  e a saída  $Q$ , iguais ao latch apresentado anteriormente. Uma saída invertida  $Q'$  muitas vezes se faz presente no dispositivo também. O flip-flop pode ser construído a partir de dois latches conectados em série, como mostrado na Figura 2.6.

O comportamento do flip-flop difere do latch em seu período de amostragem. Ao passo que o latch permanece transparente por um período de tempo, o flip-flop realiza a amostragem do sinal nas bordas de subida ou descida do *clock*.

Tabela 3 – Tabela verdade do flip-flop sensível à borda de subida

CK	D	Q
0	X	Q
$\uparrow$	0	0
$\uparrow$	1	1
1	X	Q

O comportamento ideal de ambos pode ser visto na Figura 2.7.

Os dispositivos reais, no entanto, possuem atrasos internos que estabelecem restrições no projeto de circuitos sequenciais e variam de acordo com a topologia interna do latch e flip-flop (TAN et al., 1986).

Figura 2.6 – Flip-flop tipo D

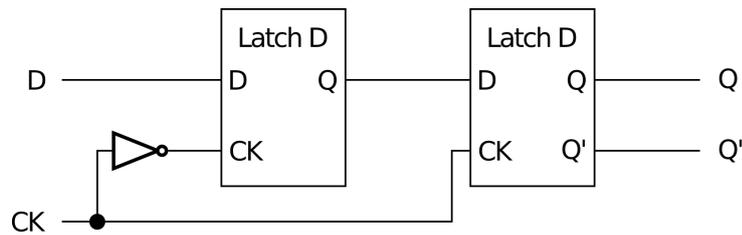
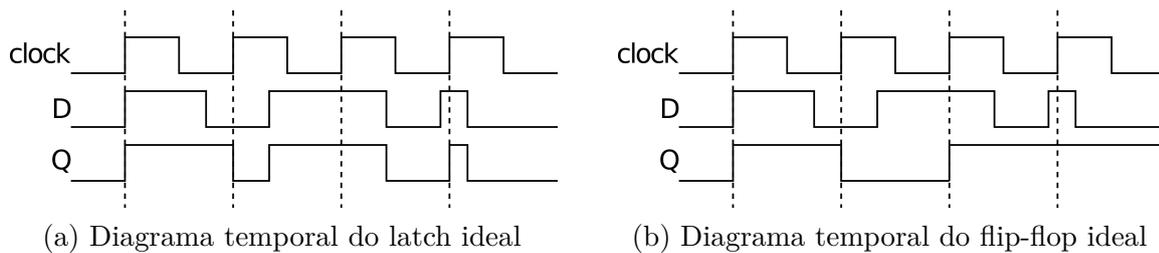


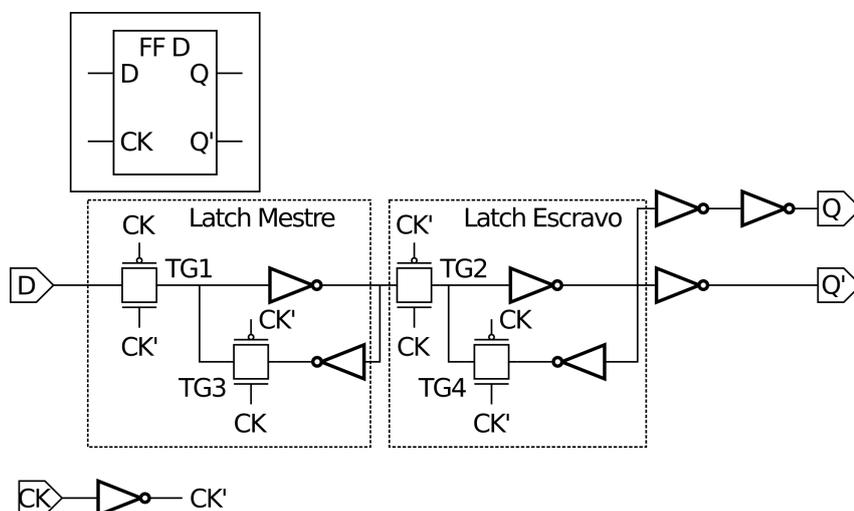
Figura 2.7 – Diagramas temporais



## 2.4.2 Elementos Sequenciais Reais

A implementação real de flip-flops compostos por um latch mestre e um escravo não é feita utilizando-se elementos combinacionais como na Figura 2.5. O uso de portas lógicas convencionais no projeto de elementos sequenciais é evitado devido ao tamanho e consumo elevados e velocidade reduzida. Flip-flops são implementados com elementos chamados portas de transmissão (do inglês *transmission gates*) que atuam como chaves controladas pelo *clock* e inversores que restauram o sinal (WESTE; HARRIS; BANERJEE, 2005). A Figura 2.8 mostra o esquemático de um flip-flop usualmente empregado em circuitos integrados.

Figura 2.8 – Esquemático de um flip-flop

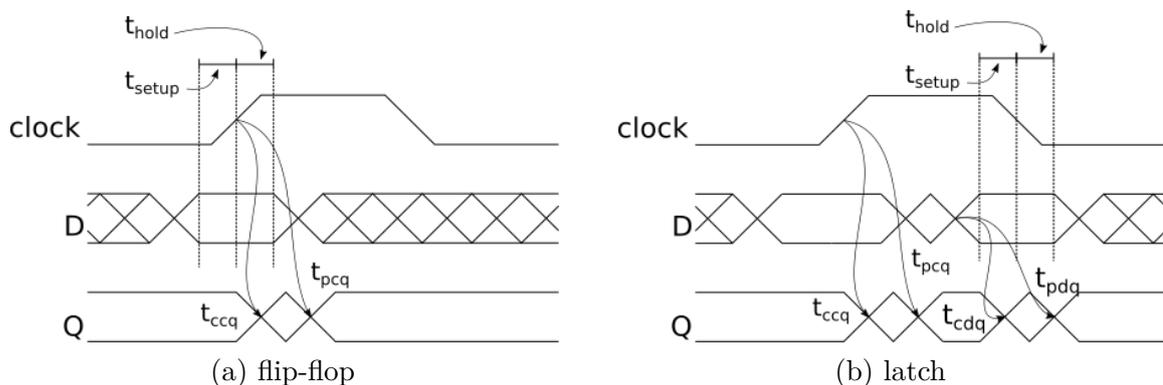


Fonte: Adaptado de Weste, Harris e Banerjee (2005)

O flip-flop da Figura 2.8 possui dois estágios de operação. Durante o nível lógico baixo do *clock* as portas de transmissão TG1 e TG4 estão transparentes enquanto TG2 e TG3 estão opacas. O latch mestre permite a entrada do sinal em D mas não há realimentação através de TG3. O latch escravo, separado do mestre por TG2, passa à saída o sinal armazenado anteriormente, mantido pela realimentação estabelecida por TG4. No nível lógico alto do *clock*, TG2 e TG3 se tornam transparente e TG1 e TG4 opacos. TG3 fecha a realimentação do latch mestre e armazena o sinal passado anteriormente através de TG1, ao passo que TG2 permite que o sinal passe à saída Q e Q'.

Os atrasos e restrições temporais de um elemento sequencial são divididos em atrasos de contaminação e propagação. Atraso de contaminação é o tempo transcorrido entre um estímulo de entrada e alguma alteração na saída do dispositivo, sem que essa mudança seja necessariamente o valor final do cômputo. Já o atraso de propagação é o tempo entre um estímulo e o valor final estável da saída (WESTE; HARRIS; BANERJEE, 2005).

Figura 2.9 – Diagrama de temporização real



A Figura 2.9 mostra os atrasos e restrições temporais de elementos sequenciais reais. O atraso de contaminação  $t_{ccq}$  é o tempo que leva para uma mudança qualquer ocorrer em Q a partir da borda de subida do *clock*;  $t_{pcq}$  é o atraso de propagação, o tempo que leva para o sinal de saída Q assentar em um valor fixo após a borda de subida do *clock*;  $t_{cdq}$  é o atraso de contaminação entre a entrada D e a saída Q;  $t_{pdq}$  é o atraso de propagação de D para Q;  $t_{setup}$  é o tempo mínimo necessário que um sinal em D deve permanecer estável corretamente amostrado;  $t_{hold}$  é o tempo mínimo que o sinal de entrada D deve permanecer estável para não comprometer a estabilidade da saída Q.

Idealmente, a lógica combinacional possui todo o ciclo de *clock* para processar o dado antes deste ser amostrado pelo elemento sequencial. Devido aos atrasos dos elementos reais, esse valor deve ser calculado a partir das restrições temporais impostas. A Equação 2.3 mostra o cálculo do período mínimo de *clock* ( $T_c$ ) necessário para que o elemento sequencial amostrasse corretamente o resultado da lógica combinacional.

$$T_c \geq t_{pdq} + t_{pd} + t_{setup} \quad (2.3)$$

O valor  $t_{pd}$  é o tempo de propagação da lógica combinacional do circuito e é estabelecido não só pelo atraso das portas lógicas, mas também pelos atrasos inseridos por fios, capacitâncias parasitas, carga dos nós de conexão entre portas (*fan out*), entre outros fatores. Intuitivamente, a Equação 2.3 pode ser expressa em função de  $t_{pd}$  para o cálculo do atraso máximo que o processamento de dado pode levar antes de ser amostrado pelo latch (TAN et al., 1986).

$$t_{pd} \leq T_c - (t_{pdq} + t_{setup}) \quad (2.4)$$

Na Equação 2.4, o termo entre parênteses é chamado de atraso sequencial. Se o reduzimos a zero (condição de latch ou flip-flop ideal), o tempo de propagação da lógica combinacional pode ser menor ou igual a todo o período de *clock* do sistema (WESTE; HARRIS; BANERJEE, 2005). No caso de latch, o tempo  $t_{pdq}$  pode ser inferior ao  $t_{pcq}$  quando o dado em D muda próximo à borda de subida do *clock*. A Equação 2.5 representa essa condição, utilizando maior atraso entre  $t_{pdq}$  e  $t_{pcq}$  (TAN et al., 1986).

$$t_{pd} \leq T_c - (\max(t_{pdq}, t_{pcq}) + t_{setup}) \quad (2.5)$$

Nos circuitos baseados em flip-flop, a lógica combinacional situa-se entre dois elementos de memória, onde toda a computação necessária é feita dentro de um único ciclo de *clock*. A Figura 2.10 mostra um diagrama temporal de um circuito com flip-flops. O primeiro flip-flop  $F1$  insere na lógica combinacional o valor para cálculo e, após um tempo  $t_{pd}$ , esse valor é disponibilizado para o segundo flip-flop  $F2$ . Se  $t_{pd}$  é muito pequeno, o resultado ficará esperando por um tempo desnecessariamente longo até a próxima borda de subida do *clock* para ser amostrado por  $F2$ . Se  $t_{pd}$  for muito grande, o resultado da lógica combinacional pode não estar pronto a tempo de ser amostrado em  $F2$  e gerar uma falha na operação do sistema. Esse comportamento de “bordas rígidas” (*hard edges*) é característico de circuitos com flip-flops (WESTE; HARRIS; BANERJEE, 2005).

Um circuito utilizando latch pode ser dividido em duas fases, como ilustrado na Figura 2.11. Ao invés de alimentar os elementos sequenciais com o *clock*, como ocorre no caso de flip-flop, os latches são controlados pelos pulsos  $\phi_1$  e seu inverso  $\phi_2$ . Quando  $\phi_1$  está em nível lógico alto, os latches  $L1$  e  $L3$  se tornam transparentes, com o primeiro inserindo um valor na lógica combinacional inicial. O pulso  $\phi_2$  torna  $L2$  transparente quando  $L1$  e  $L3$  estão opacos ( $\phi_1$  em nível baixo). O resultado da primeira lógica é então passado à segunda (BERNSTEIN et al., 1998).

Figura 2.10 – Diagrama temporal de um circuito com flip-flops

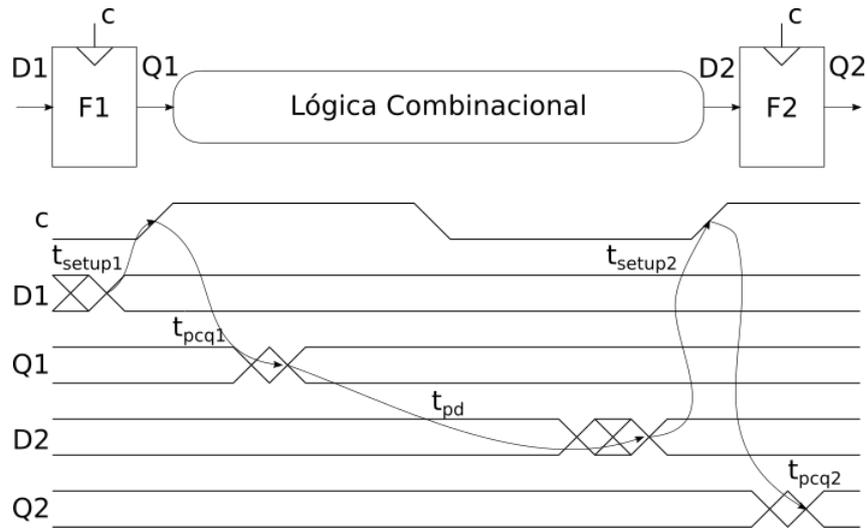
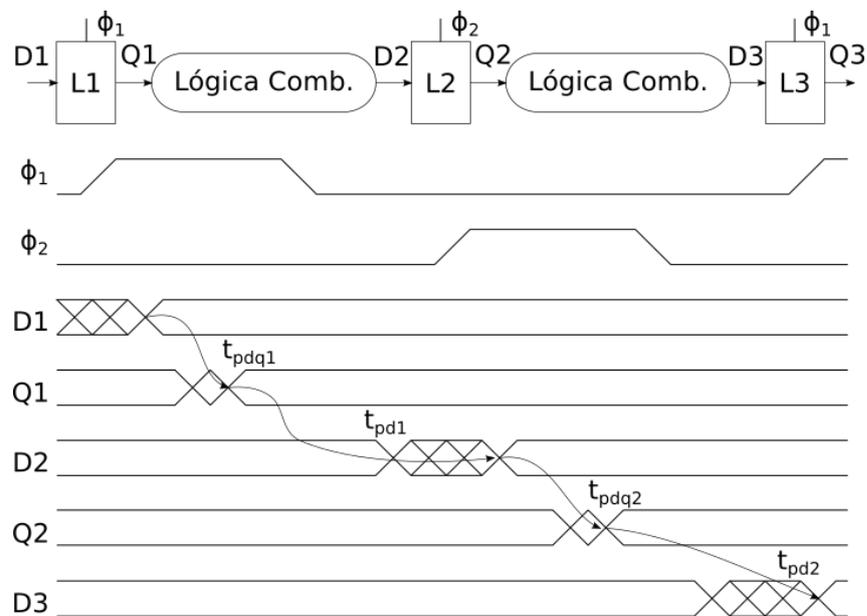


Figura 2.11 – Diagrama temporal de um circuito com duas fases

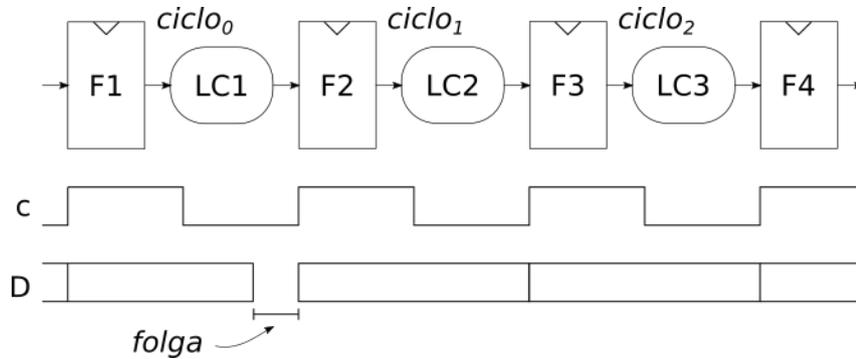


O resultado processado pela primeira lógica combinacional não precisa ficar pronto, necessariamente, na primeira fase de circuito. Já que o latch  $L2$  ficará transparente ao longo da segunda fase inteira, o resultado da primeira lógica combinacional pode utilizar tempo da segunda fase. Essa realocação de tempo chama-se empréstimo temporal (*time borrowing*) e para que o sistema emita um valor correto ao final de ciclo, o circuito que sucede o latch  $L2$  deve receber o resultado atrasado e processá-lo antes do final da segunda fase.

Empréstimo temporal também pode ser implementado com flip-flops. Como o funcionamento desses dispositivos é baseado nas bordas de *clock* ao invés de níveis lógicos, o empréstimo temporal depende de circuitos adicionais e sinais de sincronismo diferentes de *clock* do sistema para funcionar. Como não existem fases em circuitos com flip-flops,

a realocação temporal deve sempre acontecer entre ciclos de *clock* (BERNSTEIN et al., 1998). A Figura 8 mostra um circuito com vários estágios em configuração tipo *pipeline*.

Figura 2.12 – Diagrama com *pipeline* e temporização dos dados



O estágio processado no *ciclo<sub>0</sub>* consome somente 80% do ciclo de *clock*, ao passo que os demais estágios utilizam totalmente seus ciclos. O tempo excedente do primeiro estágio, chamado *folga* (do inglês *slack*), pode ser realocado para o segundo de modo a descongestionar a lógica seguinte. Uma forma de se executar essa realocação é através da alimentação de flip-flop *F2* com um *clock* adiantado em relação aos demais. Dessa forma, *F2* amostrará o resultado da primeira lógica combinacional mais cedo, reduzindo a *folga* nesse estágio e inserindo-a no próximo (BERNSTEIN et al., 1998).

O descongestionamento do segundo estágio (e possivelmente do terceiro) é um atrativo para a confiabilidade do sistema. Se os estágios possuem mais tempo para processar seus resultados, menor a chance de ocorrer uma falha por perda da borda do *clock*.

### 2.4.3 Falha de Temporização

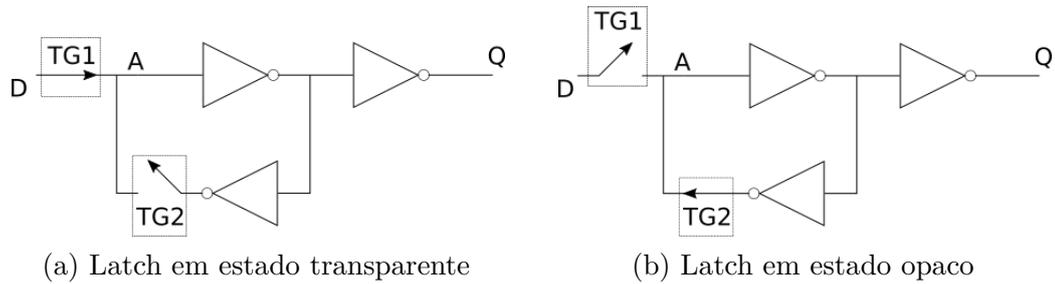
Como visto anteriormente, flip-flops e latches possuem duas restrições temporais além dos atrasos normais de operação. Uma exige que o sinal em D esteja estável antes da borda de *clock* por um tempo  $t_{setup}$ , enquanto a outra requer que o sinal permaneça estável até se passar um determinado tempo  $t_{hold}$  após a borda. Se D muda dentro dessa janela de tempo, o elemento sequencial pode tornar-se metaestável.

A metaestabilidade caracteriza-se pelo aumento considerável do tempo de propagação do elemento sequencial e da incerteza do valor que este irá emitir na saída.

Latches e flip-flops são elementos biestáveis, isso é, possuem dois estados possíveis de saída, “0” e “1”. Quando uma mudança no valor de entrada ocorre muito próxima a borda de subida do *clock*, violando a janela definida por “ $t_{setup} + t_{hold}$ ”, o elemento adquire tensões internas que não são logicamente definidas.

Na Figura 2.13, o ponto A conserva o valor lógico armazenado através da realimentação criada entre os inversores. As portas de transmissão TG1 e TG2 controlam os estados

Figura 2.13 – Estados do latch



do dispositivo. Quando o *clock* está em nível lógico alto, TG1 está conduzindo e TG2 está aberto como representado na Figura 2.13a. Nesse estado, variações em D aparecem na saída Q. O latch está transparente. Quando o *clock* está em nível lógico baixo, TG1 abre enquanto TG2 conduz e estabelece a realimentação (Figura 2.13b). Agora, nenhuma alteração em D afeta a saída e o último valor que D possuía antes da borda de descida do *clock* está retido em A.

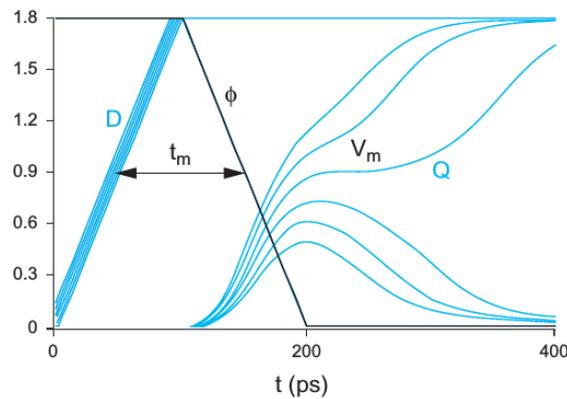
Se D muda enquanto as portas de transmissão estão sendo chaveadas pelo *clock*, o ponto A pode não estabelecer tensão suficiente para ter um nível lógico definido, em torno da metade da tensão de alimentação ( $V_m$ ). Quando as portas de transmissão se estabilizarem no novo nível lógico de *clock*, a tensão indefinida em A irá transicionar para um valor baseado na tensão do nó e no ruído interno do dispositivo. Essa transição possui um tempo de assentamento baseado nas capacitâncias e resistências em A e resulta em um  $t_{pdq}$  maior do que o observado na operação normal do elemento (WESTE; HARRIS; BANERJEE, 2005).

Na Figura 2.14a, vários sinais na entrada “D”, em diferentes instantes, são representados incrementalmente mais próximos da borda de descida do *clock*  $\phi$  que sincroniza um latch. O sinal “Q” de saída apresenta diferentes valores finais e atrasos, demorando cada vez mais para decidir pelo nível lógico final. Na Figura 2.14b o tempo  $t_{DC}$  é a diferença entre o instante de transição do sinal na entrada “D” e a borda de descida de  $\phi$ . A janela de metaestabilidade do elemento é representada por um tempo  $t_m$ . A medida que a diferença  $t_{DC} - t_m$  diminui, isso é, que o sinal “D” muda mais próximo da janela  $t_m$ , maior é o atraso de propagação do latch. O eixo vertical representa o atraso de propagação da entrada “D” para a saída “Q”.

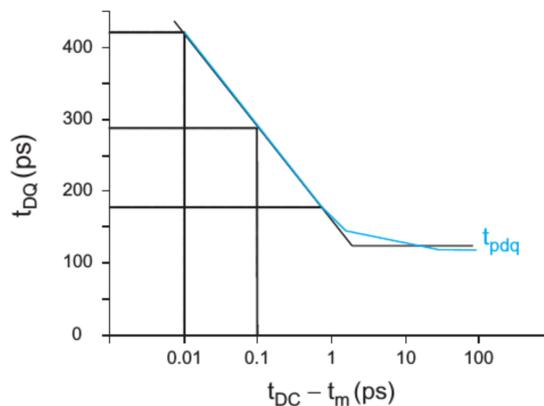
## 2.5 Fluxo de Projeto de Circuitos Digitais

O aumento da complexidade dos circuitos integrados, trazido pelo crescente número de transistores em um único chip, levou ao desenvolvimento de ferramentas de projeto eletrônico automático, do inglês *Electronic Design Automation* (EDA). Avanços em miniaturização geraram circuitos mais complexos e que requisitavam ferramentas mais

Figura 2.14 – Atraso gerado pela metaestabilidade em um elemento sequencial



(a) Saídas Q para diferentes entradas em D

(b) Variação do atraso de propagação em função de  $t_{DC} - t_m$ 

Fonte: (WESTE; HARRIS; BANERJEE, 2005)

elaboradas em seu desenvolvimento para manter os projetos em tempo hábil. Placas de circuito impresso (PCB), por exemplo, contam com algumas dezenas de componentes discretos em sua estrutura, ao passo que CIs possuíam 40.000 transistores em 1984 e cerca de 40.000.000 em 2000.

Assim como a padronização de letras e caracteres de escrita permitiram a revolução da imprensa, a padronização de células lógicas fez o mesmo à automação trazida pelas ferramentas de EDA. Uma biblioteca de células padrão utiliza uma altura comum para todos os elementos lógicos que a compõem. Assim, as ferramentas de posicionamento conseguem encaixar um elemento ao outro de maneira alinhada e o roteamento é conduzido em seguida de forma automática, exigindo pouca interação do projetista (LAVAGNO; SCHEFFER; MARTIN, 2006).

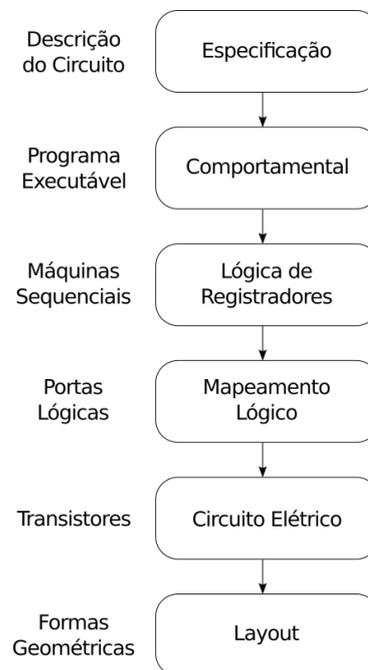
Projetos de CIs precisam ser refinados ao longo de várias camadas de detalhamento. Começando pela especificação do que o chip deve fazer, o projetista deve montar uma arquitetura capaz de cumprir as funcionalidades necessárias, expandi-la para um mapa

lógico e em seguida gerar o layout do chip. Ao longo dessas etapas, custos e limitações são estabelecidos como velocidade de operação, consumo e área.

Duas abordagens para projetos de CIs são usualmente empregadas: Projeto por hierarquia e projeto por abstração. No primeiro, o circuito é descrito através de uma lista com blocos funcionais simples. Cada bloco é quebrado em funções mais básicas e assim sucessivamente. Esse aprofundamento na árvore hierárquica continua até que se tenha funções simples o suficiente para serem descritas manualmente. Essa técnica é muito útil em circuitos onde algumas estruturas tendem a se repetir, como somadores.

No projeto por abstração, múltiplas camadas são empregadas para eliminar detalhes desnecessários em uma dada etapa do projeto. Na camada de nível lógico, por exemplo, são empregadas otimizações booleanas que não necessitam de parâmetros elétricos dos transistores. Somente as funções lógicas das células e um modelo simplificado de atraso são necessários para garantir que o circuito possua a função e temporização esperadas. Essas abstrações melhoram o tempo de projeto, com simulações e processamentos mais rápidos, gerando também maior foco dos profissionais em atributos importantes da camada em questão.

Figura 2.15 – Fluxo de projeto de circuitos integrados digitais



O diagrama da Figura 2.15 mostra as várias camadas utilizadas em um projeto por abstração. Cada etapa acrescenta detalhes ao circuito em relação a sua etapa superior enquanto abstrai informações desnecessárias contidas na inferior. As camadas são:

- **Especificação:** O cliente especifica as funcionalidades do chip. Requerimentos como velocidade mínima de operação e consumo são estabelecidos. A especificação é quase

sempre incompleta, sendo um conjunto de expectativas do chip e não um projeto.

- **Comportamental:** É a descrição comportamental do chip, com muito mais precisão do que a especificação. Usualmente essa etapa consiste na elaboração de um executável com o comportamento esperado do circuito final.
- **Lógica de Registradores:** Nessa etapa o circuito é construído em uma linguagem específica de descrição de *hardware*, como Verilog ou VHDL, usando elementos de memória abstratos e lógica booleana. Somente uma estimativa de temporização e de área podem ser feitas.
- **Mapeamento Lógico:** O sistema é mapeado utilizando portas lógicas, latches e flip-flops. A estimativa de área e atraso se tornam mais precisas.
- **Circuito Elétrico:** Um esquemático do chip é criado já em termos de transistores e elementos passivos.
- **Layout:** Etapa final para fabricação. O chip é transformado em formas geométricas que serão utilizadas pelas máscaras litográficas no processo de fabricação. Resistências e capacitâncias parasitas se tornam parâmetros calculáveis nessa etapa e simulações mais precisas podem ser feitas.

Ferramentas de síntese são utilizadas para traduzir uma especificação ou linguagem de alto nível em uma lista de conexões elétricas ou lógicas, chamada *netlist*. O fluxo de projeto, no entanto, não é unidirecional. Sempre que uma nova etapa é alcançada, ferramentas especiais de verificação são utilizadas para avaliar se o sistema cumpre as restrições de projeto estabelecidas nas camadas superiores (WOLF, 2008).

O projeto de circuitos integrados assistido por computador através de *softwares* especializados garante um menor tempo para mercado e permite a elaboração de sistemas complexos. Mesmo utilizando muita automação, essas ferramentas exigem um bom conhecimento do projetista, que precisa verificar cada resultado emitido pelo programa em busca de falhas e discrepâncias. Além disso, o projeto de circuitos realiza muitas trocas entre custos e limitações. Para que um chip seja rápido, a sua área e consumo precisam ser maiores, por exemplo. O projeto de chips é dominado por limitações e interesses conflitantes.

## 2.6 Estado da Arte

As técnicas tradicionais de redundâncias vistas na Seção 2.3 geram altos custos adicionais na forma de aumento na área, consumo e queda de desempenho. Além dessas redundâncias, é costumeira a adoção de margens de segurança para a operação confiável do

sistema. Circuitos comerciais tendem a operar em tensões mais elevadas e frequências mais baixas do que necessitam. Essas margens aumentam a tolerância à variações inseridas pelo processo de fabricação e pela degradação, mas são abordagens considerados pessimistas (FIROUZI et al., 2015).

Entre as possíveis falhas que podem afetar um circuito digital, as resultantes por problemas de temporização estão entre as mais comuns e críticas em novas tecnologias (WANG et al., 2007). A detecção dessas falhas se torna ainda mais problemática em circuitos complexos, onde a quantidade de caminhos lógicos críticos se torna grande demais para o monitoramento individual de cada um. Os caminhos críticos possuem a maior carga lógica do sistema e por isso são os principais delimitadores da frequência de operação do circuito.

Devido ao aumento das demandas por desempenho e confiabilidade, trabalhos recentes são direcionados a métodos adaptativos capazes de lidar com erros em tempo de operação do circuito, focando na eliminação das margens pessimistas.

Uma técnica comumente abordada é o Escalonamento Dinâmico de Tensão e Frequência (EDTF). O EDTF muda os parâmetros do sistema baseado no grau de atividade dos circuitos, reduzindo consumo e desgaste quando a carga de trabalho é pequena e aumentando o desempenho quando a carga for grande (BURD et al., 2000). Outros trabalhos propõem formas mais avançadas de escalonamento desses parâmetros, como a utilização de tabelas do tipo *look-up* e circuitos adicionais para o rápido chaveamento entre diferentes frequências de *clock* e tensões (TSCHANZ et al., 2007).

Muitos trabalhos abordam técnicas do tipo canários. Nelas, caminhos lógicos com temporização similar aos críticos do sistema são inseridos e monitorados em busca de sinais de degradação do circuito. A informação obtida desses caminhos são utilizadas para compensações em tensão e frequência de operação ou para diagnóstico do sistema (DRAKE et al., 2007; NAKAI et al., 2005).

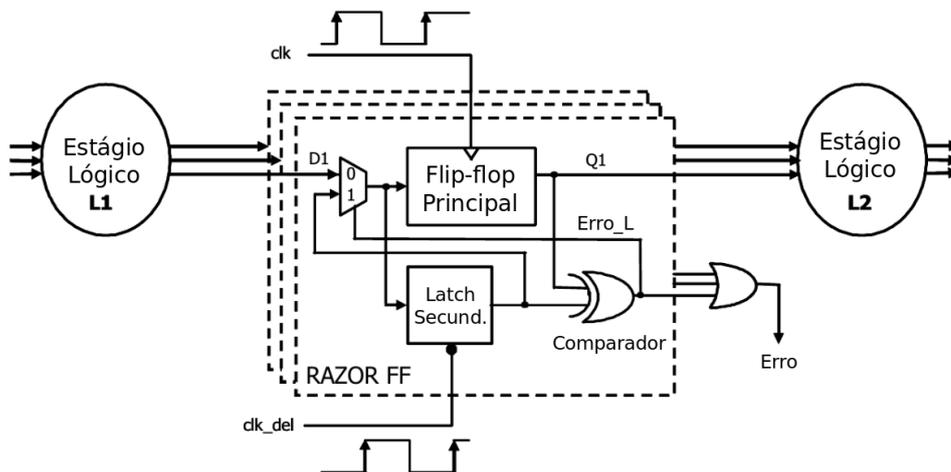
De especial interesse para o presente trabalho são as técnicas que lidam com a detecção e correção de erros *in situ* através da inserção de elementos sequenciais especiais. Esses elementos são flip-flops e latches alterados para detectar mudanças de sinais atrasados em relação ao *clock* do sistema e efetuar algum tipo de recuperação da máquina para seu estado normal de operação. Esse comportamento é chamado de mascaramento de erros por impedir que esses se propaguem ao nível de usuário e produzam falhas maiores. Esses métodos são propostos como formas de se reduzir as margens de operação do circuito, fazendo-o operar em frequências mais altas ou tensões mais baixas do que o usual.

### 2.6.1 Mascaramento de Erros Com Elementos Sequenciais

Como visto na Seção 2.4, um erro temporal ocorre quando o resultado de uma lógica combinacional falha em ser amostrado pelo elemento sequencial do circuito. Ernst et al. (2003) propõe um elemento de memória capaz de detectar dados que chegam atrasados em sua entrada, chamado RAZOR.

O RAZOR é composto por um flip-flop principal e um latch secundário, ou sombra. Ambos possuem suas entradas D conectadas a um multiplexador controlado por uma porta lógica ou-exclusivo (XOR). A XOR emite o sinal de erro captado pelo sistema e controla o multiplexador que chaveia a entrada do flip-flop principal entre o caminho lógico ou a saída do latch secundário. O diagrama do RAZOR pode ser visto na Figura 2.16.

Figura 2.16 – Diagrama do flip-flop RAZOR

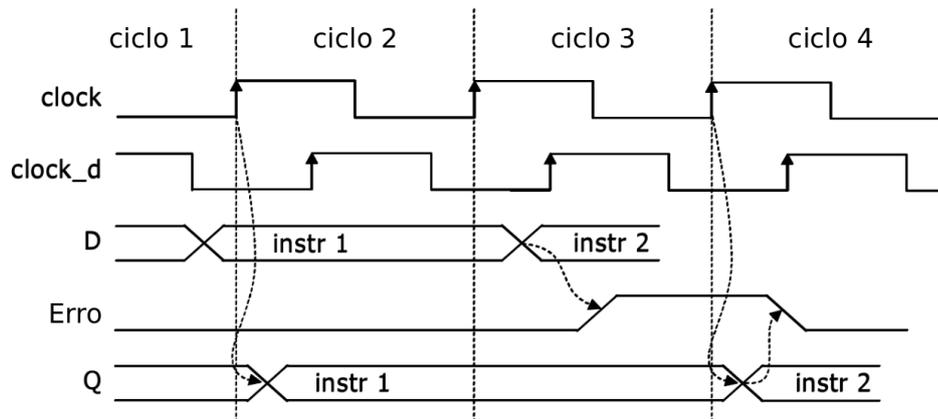


Fonte: Adaptado de Ernst et al. (2003).

O flip-flop principal é sincronizado pelo *clock* do sistema, de modo a amostrar sua entrada no tempo normal de operação do circuito. O latch secundário, no entanto, possui um *clock* atrasado, amostrando a entrada do dispositivo um tempo  $t_d$  após a borda de amostragem do *clock* normal. Em uma operação sem falhas, o RAZOR funciona como um flip-flop normal, com o dado conectado ao seu pino de entrada D sendo amostrado nas bordas de subida do *clock* e o latch secundário amostrando o mesmo dado após  $t_d$ . Como ambos enxergam um mesmo valor, a XOR terá uma saída 0, sinalizando que não há erro temporal. Se o dado de entrada do RAZOR chegar após a borda de amostragem do *clock* mas dentro da janela de amostragem do latch secundário, o flip-flop principal falhará em amostrar o dado, ao passo que o latch não. Como ambos elementos sequenciais possuem valores diferentes, a XOR sinalizará um erro temporal ao sistema. A Figura 2.17 mostra um diagrama temporal exemplificando a operação do RAZOR.

Na borda de subida entre ciclos 1 e 2 da Figura 2.17 o resultado da primeira instrução (instr 1) é corretamente amostrado no pino de entrada “D” e passado para “Q”.

Figura 2.17 – Diagrama temporal da técnica RAZOR



Fonte: Adaptado de [Ernst et al. \(2003\)](#)

Entre os ciclos 2 e 3, no entanto, o valor da instrução 2 (instr 2) muda após a borda de subida do *clock* do sistema, mas antes do *clock* atrasado do latch secundário. Ambos elementos sequenciais possuem valores diferentes, resultando no sinal de erro emitido pela XOR.

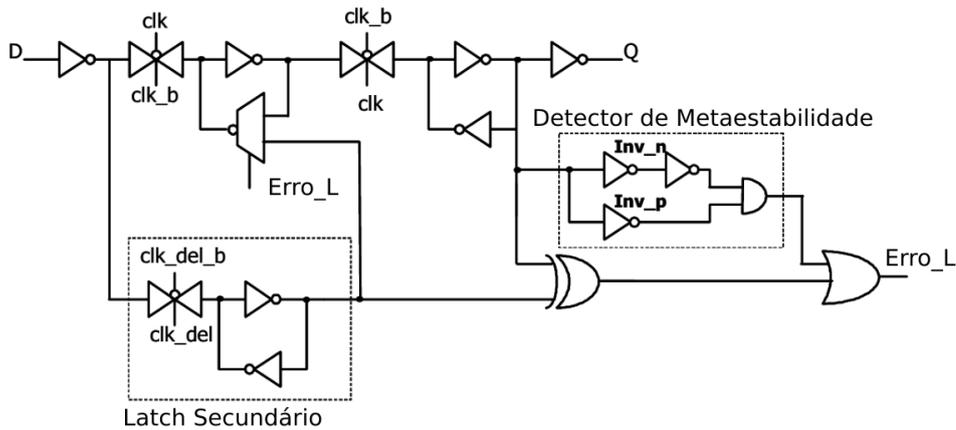
A técnica também inclui alterações na arquitetura da máquina de modo que quando um erro temporal é sinalizado, o *pipeline* é suspenso para que o valor correto do dado (retido pelo latch secundário) seja carregado no flip-flop principal. Dessa forma, a máquina retorna à sua operação normal após um ciclo de *clock* utilizado para a correção do valor.

O RAZOR foi desenvolvido com o intuito de possibilitar um circuito sequencial com *pipeline* linear a funcionar em sua tensão e frequências críticas, reduzindo ou excluindo por completo as margens de segurança usualmente empregadas. Esses parâmetros devem ser ajustados em conjunto com a taxa de erros de temporização que o circuito gera, tendo em vista que uma quantidade alta de erros produz maior consumo devido aos ciclos extras de recuperação do RAZOR.

A detecção do erro temporal requer que o flip-flop principal falhe em amostrar o dado de entrada. Isso ocorre quando o valor de entrada muda dentro da janela definida pelo  $t_{setup}$  do elemento sequencial, podendo levar o flip-flop principal à metaestabilidade. A metaestabilidade em si não é detectável naturalmente pela topologia do RAZOR, levando à necessidade de um circuito detector. O diagrama do RAZOR com detector de metaestabilidade pode ser visto na Figura 2.18.

Os inversores que antecedem a porta lógica E (AND) possuem diferentes tensões de limiar. Se o nó de realimentação do latch escravo encontra-se em uma tensão sem nível lógico definido (nas proximidades de  $V_{dd}/2$ , condição da metaestabilidade) a AND elevará a saída para nível lógico 1, alertando o sistema sobre um erro no flip-flop.

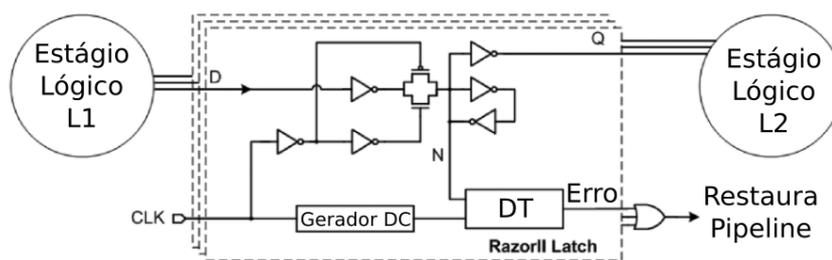
Figura 2.18 – Flip-flop RAZOR com detector de metaestabilidade



Fonte: Adaptado de Ernst et al. (2003)

A evolução do RAZOR, o RAZORII, trata de alguns dos principais problemas de seu antecessor. As modificações de arquiteturas realizadas pelo RAZOR impossibilitam sua aplicação em processadores de maior desempenho e que usufruem de *clocks* mais rápidos. O RAZORII remedia este problema utilizando a microarquitetura da própria máquina em seu favor. Para isso, ele utiliza rotinas de *replay* de instruções naturais da arquitetura. Dessa forma o RAZORII não necessita realizar mudanças na estrutura geral da máquina, tornando-o uma solução mais abrangente (DAS et al., 2009). A estrutura do RAZORII pode ser vista na Figura 2.19.

Figura 2.19 – Diagrama funcional do RAZORII

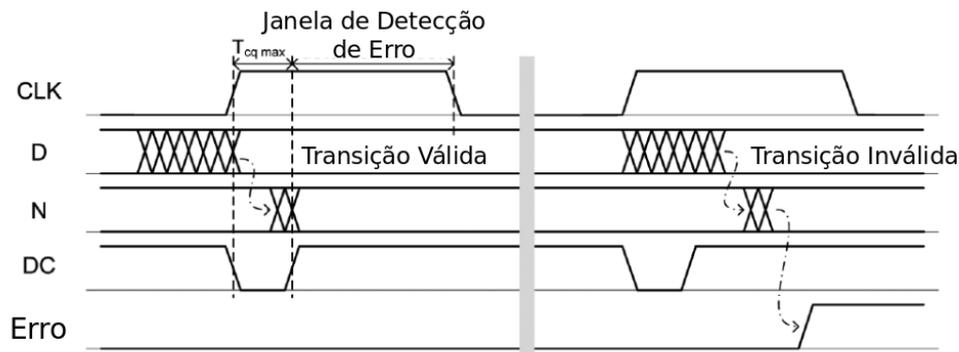


Fonte: Adaptado de Das et al. (2009).

O RAZORII é composto por um latch principal, transparente em nível lógico alto, e um gerador de pulso de detecção (DC) que controla um detector de transição (DT). O latch principal realiza a amostragem do dado de entrada na borda de subida do *clock* do sistema. O pulso de detecção de falha estabelece um período de tempo onde é proibida qualquer mudança no valor de entrada do dispositivo. Caso haja uma transição de sinal nessa janela, o DT acionará o pino de erro do circuito e iniciará o processo de *replay* da instrução.

A forma como o RAZORII funciona tem a vantagem adicional de detectar erros transientes na lógica combinacional. Qualquer oscilação de sinal que se propague até o RAZORII durante o pulso de detecção, ativará o sinal de erro do dispositivo (DAS et al., 2009). A Figura 2.20 mostra um diagrama temporal do RAZORII.

Figura 2.20 – Diagrama temporal do RAZORII



Fonte: Adaptado de Das et al. (2009).

Apesar de não fazer alterações significativas na arquitetura da máquina, o RAZORII depende da disponibilidade de instruções de *replay*. Essa limitação é remediada pelo *Bubble RAZOR*.

Ao contrário dos antecessores, o *Bubble RAZOR* emprega uma topologia de circuitos baseados em latches de duas fases, como visto na Seção 2.4. Duas vantagens surgem com essa abordagem: Elimina a dependência nos tempos mínimos e nas janelas de detecção e permite correção das falhas de forma independente da arquitetura. A primeira vantagem surge devido a estrutura de circuitos sequenciais divididos em duas fases devidamente balanceadas. Quando um latch se torna transparente, todas as computações do estágio anterior já foram concluídas, impossibilitando a ocorrência de sinais chegando muito cedo no próximo estágio. Essa característica também permite ao *Bubble RAZOR* utilizar uma fase inteira para correção de falhas (FOJTIK et al., 2013).

No *Bubble RAZOR* o erro é detectado quando o sinal de uma fase é recebido tardiamente na seguinte. O latch que detectou o erro emite uma bolha para os latches vizinhos e esses a propagam adiante. Essa bolha suspende o processamento de todos os latches por onde passam, permitindo que o estágio onde o erro foi detectado tenha tempo de processar a sua lógica e retornar à operação normal.

Até o momento, as técnicas apresentadas suspendem o funcionamento da máquina para realizar a recuperação de um erro temporal. O TIMBER lida com violações temporais sem a necessidade de gerar tais suspensões.

O TIMBER utiliza empréstimo temporal dos estágios seguintes do *pipeline*, permitindo que a lógica onde o erro foi produzido tenha mais tempo para produzir o resultado

final. Esse funcionamento é similar ao de circuitos com latches em duas fases, onde uma fase utiliza o tempo da seguinte para terminar a sua computação. Esse empréstimo pode resultar em uma propagação do erro temporal para o estágio seguinte. Nesse caso, o TIMBER permite que empréstimos temporais ocorram sucessivamente enquanto reduz a frequência do circuito até que não haja mais violações temporais.

A técnica estabelece uma janela temporal após a borda de subida do *clock* para a detecção e o mascaramento do erro, especificada durante o projeto do circuito. A janela é dividida em um intervalo de empréstimo temporal e outro para detecção de erro. Se uma violação não se propaga de um estágio para o outro, o empréstimo ocorre sem alteração do estado da máquina, utilizando-se o intervalo de empréstimo. Caso o erro se propague para os estágios seguintes, a violação ocorrerá dentro do intervalo de detecção e um sinal é emitido para a máquina, forçando a redução do *clock* para impedir que o erro se propague indefinidamente pelo *pipeline*.

A técnica propõe tanto um flip-flop quanto latch adaptados para o TIMBER. O primeiro realiza empréstimos temporais em unidades discretas de tempo, conservando as propriedades de borda dos flip-flops convencionais do tipo mestre-escravo, enquanto o segundo trabalha com alocação e detecção continuamente (CHOUDHURY et al., 2010).

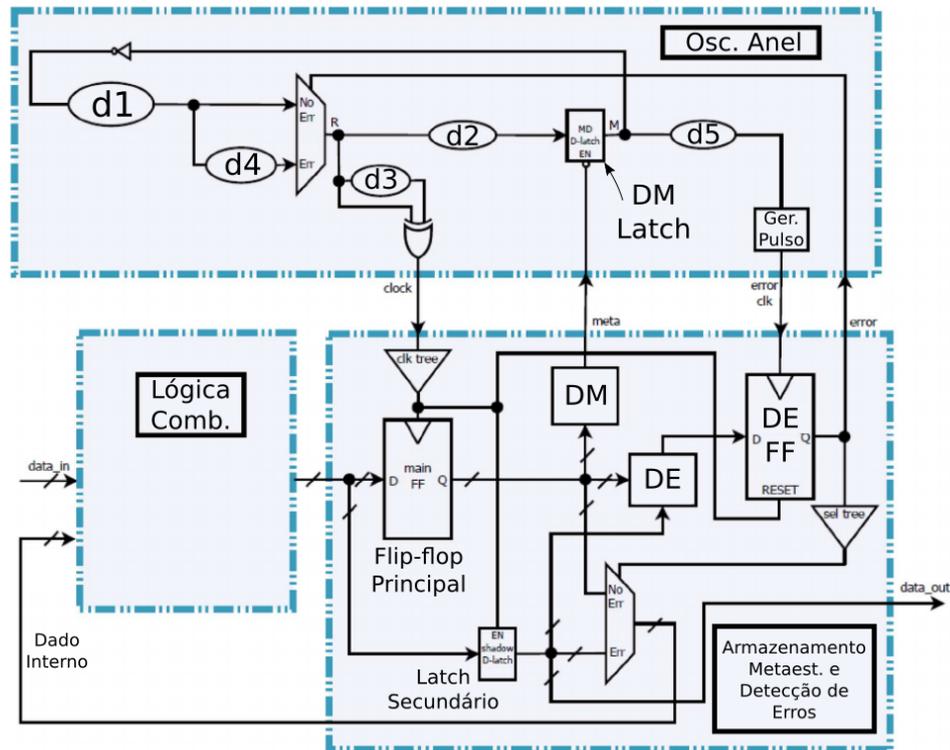
Apesar das diferentes técnicas propostas lidarem, em algum nível, com a metaestabilidade, nenhum deles prevê que ela ocorra no sinal de aviso de erro que os dispositivos emitem. Essas abordagens necessitam que esse sinal seja corretamente interpretado para que a máquina efetue um processo de recuperação do erro. Contudo, se o sinal de aviso se torna metaestável sua imprevisibilidade pode ser propagada pelo sistema, impedindo que a restauração ocorra. Para lidar com esse problema o trabalho de Cannizzaro et al. (2015) propõe o SafeRAZOR.

O SafeRAZOR é utilizado em sistemas Globalmente-Assíncronos Localmente-Síncronos (GALS), onde o elemento sequencial proposto realiza o controle de frequência da região síncrona em que se encontra. a Figura 16 mostra o diagrama de blocos do SafeRAZOR.

Durante a operação normal do módulo, o oscilador em anel utiliza os atrasos de  $d_1$ ,  $d_2$  e  $d_3$  para gerar o *clock* do circuito. A soma desses três atrasos deve ser maior do que o atraso dos caminhos críticos conectados aos SafeRAZORs. Todos os multiplexadores permanecem na posição “sem erro”, nessa situação, e o valor do latch sombra é passado à saída de dados do elemento, enquanto a saída do flip-flop principal é realimentada para a lógica combinacional. O latch no oscilador em anel está sempre transparente nesse modo.

Se o flip-flop principal se torna metaestável, o detector de metaestabilidade (MD) fecha o latch do oscilador em anel (DM Latch), suspendendo o *clock* do circuito até que a metaestabilidade seja resolvida.

Figura 2.21 – Diagrama funcional do SafeRAZOR



Fonte: Adaptado de Cannizzaro et al. (2015).

Quando um erro temporal é detectado pelo bloco DE, o flip-flop de detecção comuta os multiplexadores da arquitetura, chaveando a realimentação da lógica combinacional para o latch sombra e inserindo o atraso  $d_4$  no oscilador em anel. Esse atraso ocasiona o aumento do período do *clock* permitindo que a lógica combinacional processe o valor corrigido do latch sombra (CANNIZZARO et al., 2015).

Através da literatura verifica-se uma forte tendência em encontrar soluções que substituam os métodos convencionais e pessimistas empregados na confiabilidade de circuitos integrados. O alto desempenho requisitado de sistemas atuais somado às margens de segurança e métodos de redundância pessimistas geram altos custos que, muitas vezes, forçam sistemas a serem menos confiáveis ou eficientes. Os trabalhos apresentados nessa seção mostram um foco claro em soluções voltadas para erros oriundos de temporização dos circuitos digitais, tentando atender às margens rígidas impostas pelas altas frequências empregadas sem comprometer a segurança de operação do sistema.



## 3 Metodologia

Este capítulo descreve as técnicas propostas para lidar com a detecção e correção de erros temporais em circuitos sequenciais denominadas Empréstimo Seletivo de Tempo (EST) e Ativamento de Caminho Alternativo (APA). Serão discutidos os métodos utilizados para implementação e validação de ambas as técnicas e suas principais diferenças com às propostas na literatura.

Tendo em vista que este trabalho foca nas técnicas que propõem elementos sequenciais como forma de detectar e corrigir erros no circuito, os principais trabalhos a serem discutidos são os apresentados na Seção 2.6.1.

Este capítulo está dividido em cinco seções. A primeira apresenta uma visão geral do problema e das técnicas encontradas na literatura. A detecção de erros temporais é apresentada em seguida. As duas técnicas propostas são apresentadas nas seções seguintes, detalhando implementação e topologia. A última seção apresenta o fluxo de inserção automático dessas técnicas.

### 3.1 Visão Geral e Motivação

O RAZOR (ERNST et al., 2003) foi um dos primeiros trabalhos a propor um elemento sequencial com capacidade de detectar e corrigir erros de temporização do circuito. Mudanças de sinal que ocorrem dentro de uma determinada margem de tempo - chamada janela de detecção -, próximo a borda de subida do *clock*, são consideradas erros de temporização e tratadas pela arquitetura da máquina. Essa abordagem depende de alterações feitas no circuito e de modificações em nível de arquitetura. A restauração dos valores corrigidos também necessita de um ciclo de *clock* para carregar o valor correto, armazenado no latch secundário. Os trabalhos sucessores ao RAZOR remediaram alguns de seus problemas, mas acabaram por introduzir outros que geram restrições em operação e aplicabilidade.

Apesar de não realizar modificações na arquitetura da máquina, o RAZORII (DAS et al., 2009) depende do *replay* de instruções, resultando no esvaziamento do *pipeline* e perda desnecessária de desempenho. Além disso, a janela de detecção de erros ocupa metade do ciclo do *clock*. Caminhos lógicos rápidos que estejam conectados ao flip-flop deverão ser atrasados. Caso contrário, mudanças de valores resultantes desses caminhos podem ser interpretadas como erros temporais. De forma similar, variações esporádicas no sinal de entrada do flip-flop devido ao chaveamento das portas lógicas (*glitch*) serão interpretadas como falhas, resultando na suspensão desnecessária do *pipeline*.

Resolvendo boa parte dos problemas de sensibilidade presentes no RAZORII, o *Bubble* RAZOR (FOJTIK et al., 2013) necessita de realizar grandes mudanças no circuito, transformando-o na topologia de latches em duas fases. Essa técnica recupera os erros temporais através de bolhas que se propagam pela máquina, atrasando a entrega dos resultados finais.

O TIMBER (CHOUDHURY et al., 2010) elimina a necessidade de suspender o *pipeline* através do empréstimo temporal. Quando uma falha temporal ocorre em um estágio, uma parte do próximo ciclo de *clock* é alocada para resolver o problema. Quando um erro se propaga por vários estágios sucessivos, a frequência da máquina é reduzida até que não hajam mais violações. Assim como no RAZORII, essa abordagem é sensível a transições espúrias que podem ocorrer dentro da janela de detecção da falha. A frequência da máquina pode ser reduzida desnecessariamente devido a *glitches* ou até mesmo erros transientes que não afetariam a operação do *pipeline*. O circuito proposto pelo TIMBER também pode gerar sinais de violação metaestáveis.

Voltando-se principalmente para a questão da metaestabilidade, o SafeRAZOR (CANNIZZARO et al., 2015) é uma técnica utilizada em circuitos do tipo GALS. Nela, o metaestabilidade resulta em suspensão do *clock* da região síncrona em que se encontra. O sinal de *clock* só é retomado quando a metaestabilidade é resolvida. Violações temporais também são tratadas com a redução da frequência na região síncrona. A grande desvantagem do SafeRAZOR é a sua limitação em aplicabilidade. Circuitos digitais síncronos de alto desempenho não podem usufruir desse sistema de controle de frequência.

As técnicas elaboradas ao longo deste trabalho tem o objetivo de atacar esses problemas encontrados na literatura, estabelecendo assim objetivos que foram buscados ao longo do desenvolvimento. Eles são:

- **Propósito Geral:** As soluções desenvolvidas devem ser compatíveis com circuitos sequenciais como um todo e não somente em aplicações e arquiteturas específicas;
- **Independente de Controle:** Como visto anteriormente, técnicas estudadas na literatura usualmente incluem suspensão da máquina ou controle de frequência em um nível de sistema. Neste trabalho, busca-se por soluções que sejam implementadas em nível lógico e que aumentem a robustez do circuito de forma transparente para o sistema;
- **Orientado a Degradação:** Algumas técnicas são apresentadas como formas de se aumentar o desempenho do sistema, possibilitando a redução de margens de frequência e tensão. Neste trabalho o foco principal é a degradação. As técnicas divisadas não tem o objetivo de substituir outros métodos mas sim acrescentar robustez junto a eles.

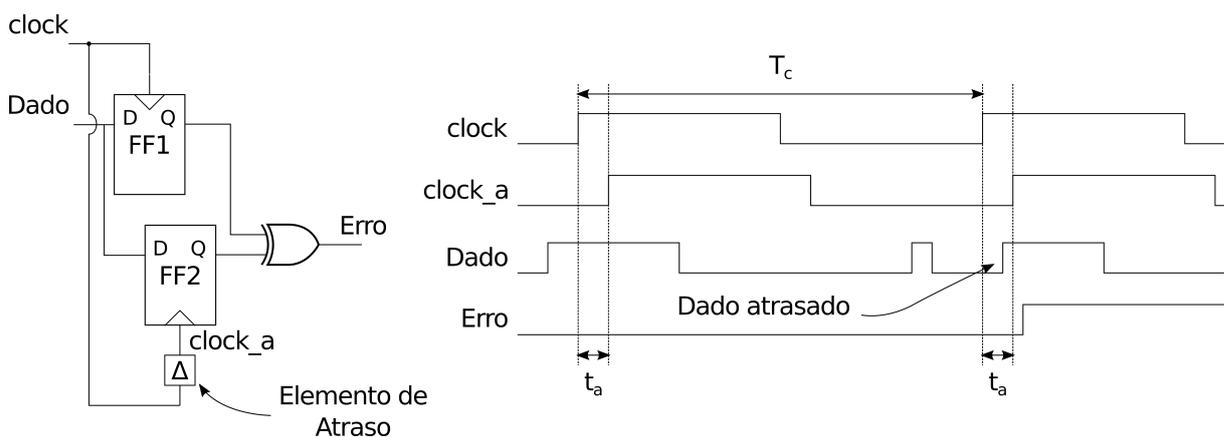
## 3.2 Detecção de Erros

Como discutido na Seção 2.4, um erro temporal - ou também chamado de erro dinâmico - ocorre quando um resultado falha em ser amostrado pelo elemento sequencial conectado ao final da lógica. Esse atraso pode ocorrer devido a variações no processo de fabricação ou aos diferentes efeitos gerados pela degradação do circuito.

A detecção do erro temporal pode ser feita através de redundância. Considere dois sinais, com níveis lógicos  $V$ , amostrados em diferentes tempos  $t_0$  e  $t_1$  por um elemento sequencial, com  $t_1 > t_0$  e ambos próximos a borda de amostragem do *clock*. O erro é detectado quando  $V_{t_0} \neq V_{t_1}$ .

A Figura 3.1 mostra um circuito capaz de detectar erros dinâmicos junto a um diagrama temporal exemplificando sua atuação. O elemento sequencial principal é um flip-flop que amostra o sinal em sua entrada D na borda de subida do *clock*. O flip-flop secundário realiza a amostragem da mesma entrada com um atraso  $t_a < T_c$ , com  $T_c$  o período do *clock*. Um circuito comparador (nesse caso uma porta lógica XOR) emitirá um sinal de erro se ambos os flip-flops amostrarem valores diferentes. Inicialmente o valor “Dado” estabiliza antes da borda de subida do primeiro ciclo de *clock*, sendo amostrado igualmente pelos dois flip-flops do circuito. Já no segundo ciclo, o valor de “Dado” chega tardiamente, entre a borda de subida do sinal de *clock* e de seu equivalente atrasado *clock\_a*. Nessa condição, FF1 falha em amostrar o sinal “Dado” ao passo que FF2 capta o novo valor. A diferença na saída de ambos os elementos sequenciais é captada pela porta XOR e um sinal de “Erro” é emitido.

Figura 3.1 – Diagrama temporal de detecção de erro



O sinal de erro pode ser utilizado para a ativação da recuperação do sistema, como visto nas diferentes técnicas propostas na literatura.

Considere agora um circuito sequencial organizado na forma de um *pipeline* composto por  $n$  estágios  $E_0, E_1, \dots, E_{n-1}$ , com cada estágio conectado ao seguinte através de elementos sequenciais formando bancos de registradores. Se o circuito opera em seu *clock*



erros, reduzindo o impacto de qualquer técnica de monitoramento em área e consumo (WANG; CHEN; TEHRANIPOOR, 2012). Essas características em circuitos sequenciais são exploradas nas técnicas desenvolvidas neste trabalho.

### 3.3 Empréstimo Seletivo Temporal

Ao final da Seção 2.4 foi discutida a possibilidade de empréstimo temporal com circuitos baseados em flip-flops. Um estágio sequencial com folga temporal pode ceder tempo ao seguinte se o flip-flop que os separa é sincronizado com um *clock* adiantado em relação aos demais. Essa solução, no entanto, exige a criação do sinal de sincronismo adiantado. Em circuitos complexos o *clock* é distribuído através de uma estrutura em forma de árvore, que cria regiões de baixa variação do sinal em seus ramos (KURD et al., 2001). O *clock* proveniente do “tronco” da árvore pode ser usado como sinal adiantado para o empréstimo temporal, transferindo tempo de estágios sequenciais menos congestionados para os críticos (BERNSTEIN et al., 1998). Essa transferência de folga também pode ser realizado em sentido contrário. Um flip-flop alimentado com um *clock* atrasado transfere tempo para o estágio conectado em sua entrada.

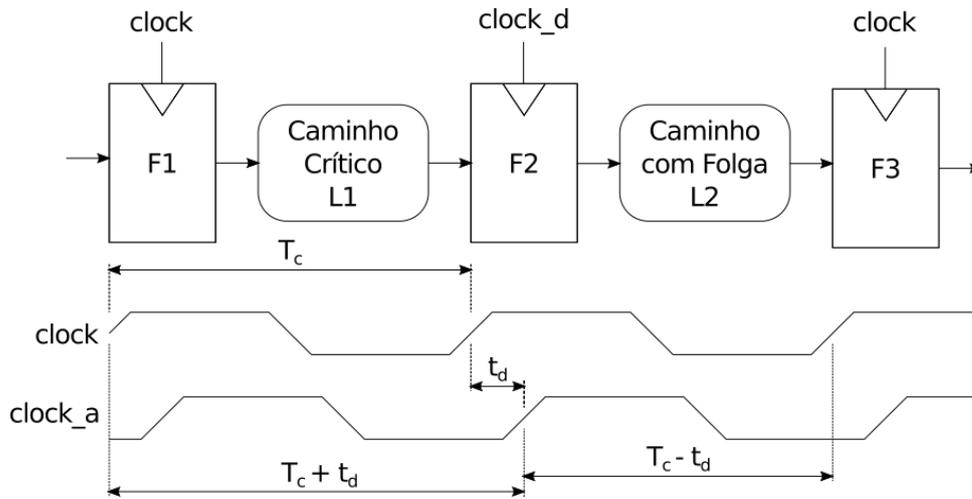
A técnica intitulada Empréstimo Seletivo Temporal (EST) utiliza esse princípio como forma de redistribuir as folgas entre os estágios do circuito. O flip-flop conectado ao final do caminho crítico tem o seu sinal de *clock* atrasado, utilizando parte da folga do estágio seguinte para aguardar qualquer dado que possa vir atrasado. A técnica visa balancear a distribuição temporal entre os caminhos e assim reduzir as chances de erros dinâmicos, repassando folga para os caminhos que necessitam mais tempo.

A detecção do erro temporal não é utilizada pelo EST, tendo em vista que o erro não é tratado especificamente. Todo sinal que transicionar dentro da janela de empréstimo temporal do EST será tratado como uma mudança normal e será amostrado pelo flip-flop sem que haja suspensão do sistema ou mudança de frequência. Dessa forma, a técnica também não realiza detecção de falhas transientes. O sinal que estiver presente na entrada do elemento sequencial, no instante da amostragem, será armazenado.

Na Figura 3.3 o caminho crítico L1 possui  $T_c + t_d$  para processamento, ao passo que L2 possui  $T_c - t_d$ . Uma parcela  $t_d$  da folga disponível em L2 foi transferida para L1. O balanceamento temporal, no entanto, não pode realizar uma transferência total da folga. Caso contrário, o caminho L2 se tornaria crítico e uma possível fonte de erros. O balanceamento é conduzido transferindo-se metade da folga disponível de um estágio para o seu anterior, como mostrado na Equação 3.1.

$$t_{balanceado} = \frac{t_{folga}}{2} \quad (3.1)$$

Figura 3.3 – Pipeline com EST



O valor de  $t_{folga}$  é calculado subtraindo-se o atraso de propagação do período de  $clock$ , levando-se em conta o valor de  $t_{setup}$ , que não pode ser violado. Assim,  $t_{folga} = t_{pd} - (T_c - t_{setup})$ . Utilizando a Equação 3.1 tem-se:

$$t_{balanceado} = \frac{t_{folga}}{2} = \frac{t_{pd} - (T_c - t_{setup})}{2} \quad (3.2)$$

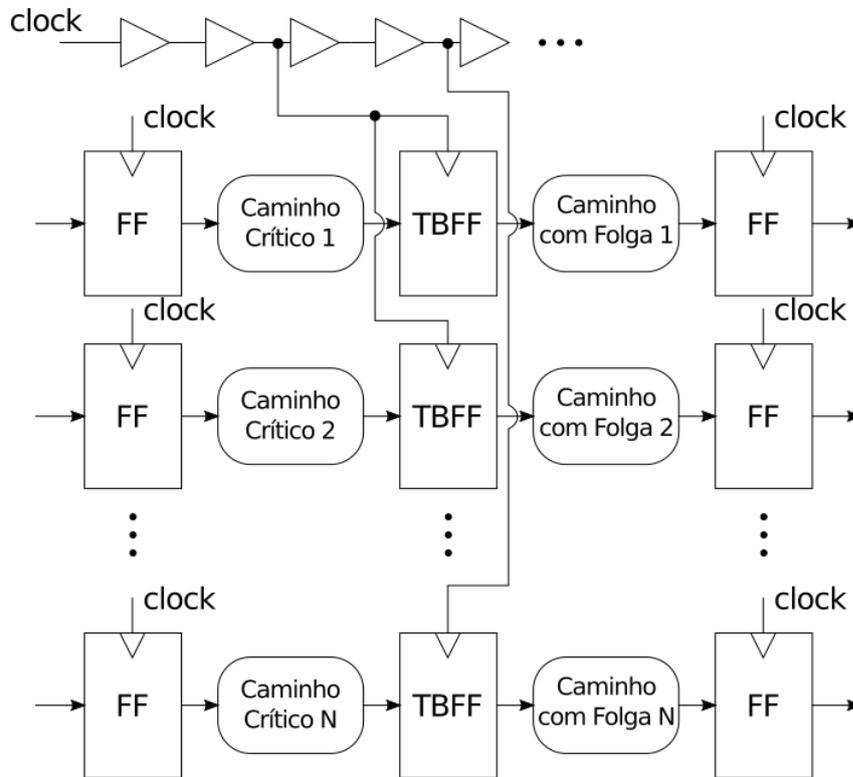
Se o valor de  $t_{folga}$  é muito pequeno, ou seja, o caminho seguinte ao crítico é tão congestionado quanto ele, a transferência de folga pode não ser possível. Essa avaliação deve ser feita durante a inserção da técnica, quando uma margem de folga mínima para balanceamento é definida. Se a margem é estabelecida em 20%, por exemplo, todo caminho seguinte aos críticos que possuem esse valor ou mais do ciclo de  $clock$  disponível poderão ter sua folga balanceada pelo EST.

A geração de um  $clock$  atrasado é mais simples do que a de um adiantado. Uma cadeia de  $buffers$  pode ser empregada para gerar sinais com diferentes atrasos, como mostra a Figura 3.4.

Cada  $buffer$  produz um sinal com mesma polarização que o  $clock$  original do sistema, atrasado em  $t_{buffer}$ . Na Figura 3.4 os diferentes atrasos gerados por essa cadeia são utilizados para alimentar os flip-flops modificados pelo EST, chamados TBFF (do inglês *Time-Borrowing Flip-Flop*), fornecendo atrasos próximos ao  $t_{balanceado}$  da equação 3.2 para cada folga disponível no estágio seguinte. O valor  $t_{balanceado}$  e a quantidade de  $buffers$  necessários para alcançá-lo é calculado durante o projeto do circuito.

Devido à característica discreta da cadeia de  $buffers$ , o tempo é realocado em degraus múltiplos de  $t_{buffer}$ . Essa granularidade limitada e a dependência do atraso a fatores como tecnologia e carga dos  $buffers$  tornam difícil prever o atraso real da cadeia quando fabricada. Variações na alimentação dos componentes também afetará o atraso, contribuindo para essa imprevisibilidade. Circuitos de atraso regulável são alternativas desejáveis à cadeia de

Figura 3.4 – Empréstimo temporal utilizando atraso de cadeia de inversores



*buffers* simples, como exemplo pode-se mencionar cadeias de inversores do tipo *current-starved*, onde o atraso produzido na cadeia é proporcional a uma tensão de referência (DALLY; POULTON, 2008). O emprego de outros circuitos de atraso junto ao EST, no entanto, não foi explorado neste trabalho.

A metaestabilidade não é um problema para essa técnica, já que o *clock* atrasado provê um maior tempo para o dado no caminho crítico se estabilizar. Um atraso consideravelmente grande na estabilização do sinal de entrada ainda poderia levar o TBFF à metaestabilidade. Nesse caso o circuito estaria operando no limite de sua vida útil, com os caminhos críticos entregando sinais próximos à amostragem do *clock* atrasado. A técnica foi desenvolvida para prover essa vida útil adicional ao circuito. Nesse caso, não fosse pela técnica, o circuito já teria falhado antes de atingir esse ponto de operação.

Outra situação que pode gerar metaestabilidade no TBFF ocorre com caminhos lógicos curtos conectados ao mesmo flip-flop que o crítico. Se uma transição de sinal gerado por um desses caminhos ocorre próximo a amostragem do flip-flop, metaestabilidade também ocorrerá no elemento sequencial ou uma mudança legítima de sinal poderá ser confundida por um erro temporal e passado adiante para o próximo estágio. Essa situação pode ser resolvida a partir da inserção de atraso em caminhos curtos, impedindo que transições ocorram na janela temporal. Essa solução permite manter o empréstimo temporal como dimensionado para o caminho crítico, conservando o benefício da técnica. O custo dessa solução será o aumento da área e consumo gerados pelas células de atraso inseridas

no caminho. Alternativamente, a janela temporal pode ser reduzida, o que resultará em menor empréstimo para o caminho crítico e um menor impacto no circuito. Essa relação custo-benefício deve ser analisada e depende da aplicação em que o sistema irá atuar.

Um problema no uso do EST surge quando um caminho crítico é sucedido por outro igualmente congestionado, não havendo folga para ser redistribuída. Nessa situação, o atraso do *clock* no elemento que interfaceia os dois caminhos resultaria em um erro temporal no estágio seguinte. Uma forma de se lidar com esse problema é através do empréstimo temporal sucessivo, onde folga é realocada de estágios mais avançados de modo similar ao TIMBER. O empréstimo temporal realizado ao longo de mais de dois estágios, no entanto, é uma solução que restringe a aplicabilidade da técnica. Nesses casos faz-se necessário um controle da máquina em nível mais alto para impedir que o erro se propague indefinidamente pelo *pipeline*. O TIMBER, como visto antes, reduz a frequência da máquina quando o empréstimo é feito ao longo de vários estágios.

O objetivo principal do EST é realizar o empréstimo temporal sem que alterações na arquitetura da máquina sejam feitas. Nesse caso, uma outra solução se faz necessária para quando um caminho crítico é sucedido por outro. Na seção a seguir é apresentada a técnica que lida com esse problema, chamada Ativação de Caminho Alternativo.

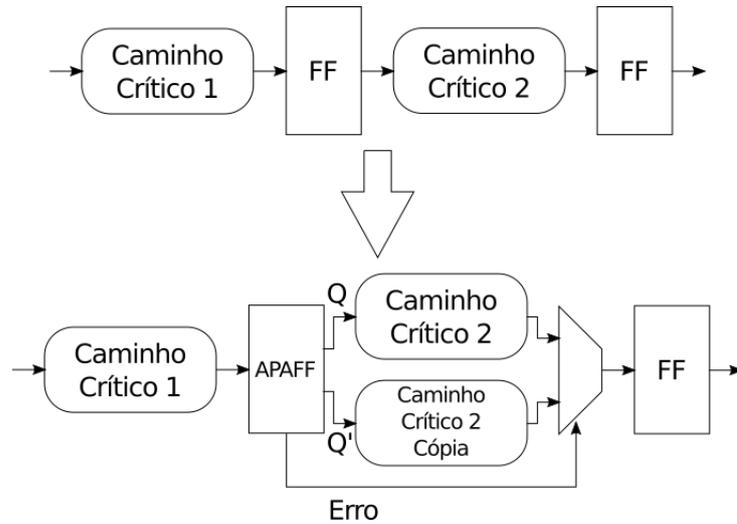
### 3.4 Ativação de Caminho Alternativo

Como visto na Seção 3.3, quando um caminho crítico é sucedido por outro igualmente congestionado não é possível realizar o empréstimo temporal. O método desenvolvido para lidar com essas situações é chamado Ativação de Caminho Alternativo (APA). A técnica utiliza uma redundância física com o objetivo de recuperar um erro dinâmico sem realocar tempo do caminho seguinte ao crítico, sendo aplicável em situações onde o empréstimo temporal não é viável.

Um elemento sequencial possui, usualmente, duas saídas: Uma saída de polarização igual a entrada, chamada  $Q$  e uma invertida  $Q'$ . Quando um erro temporal ocorre, o valor que foi passado pelo flip-flop através da saída  $Q$  está errado. Dada a natureza binária dos sistemas computacionais, é seguro afirmar que se a saída  $Q$  está errada, então a  $Q'$  está certa, já que ela é a única alternativa ao erro. Se um erro temporal é detectado e a saída  $Q$  do flip-flop emitiu o valor lógico “1”, então o valor correto é “0” e vice-versa. A técnica APA explora essa característica para restaurar o valor correto sem que o tempo do estágio seguinte seja utilizado.

Considere um caminho crítico CC1 conectado através de um flip-flop ao segundo estágio composto pelo caminho crítico CC2. Dado que CC1 só pode emitir um entre os dois níveis lógicos adiante, uma redundância física pode ser inserida de modo a processar os dois possíveis valores emitidos por CC1, como mostra a Figura 3.5.

Figura 3.5 – Diagrama de funcionamento do APA



O APA utiliza uma cópia do caminho lógico CC2 e o conecta à saída  $Q'$  do flip-flop de interface com CC1. Um circuito de detecção de erro temporal emite um sinal que controla um multiplexador ao final da duplicação. Durante a operação normal do sistema, o sinal de erro está em nível baixo e CC2 está selecionado no multiplexador, passando adiante o processamento da saída  $Q$ . Quando um erro temporal é detectado, o multiplexador seleciona a cópia de CC2. Essa emite o resultado do processamento de  $Q'$  para o próximo estágio. O caminho original e sua cópia compõem um domínio duplicado que, para o resto do circuito, possui exatamente a mesma função lógica que somente o caminho original.

O sinal de erro gerado pelo flip-flop do APA (APAFF) comuta diretamente o multiplexador, garantindo o ganho de tempo para o caminho crítico. Esse novo elemento sequencial é composto por dois flip-flops, um principal responsável pela operação quando não há erro temporal e um secundário utilizado na detecção do erro. A estrutura do APAFF é apresentada mais adiante na Seção 3.4.1.

O tempo disponível para a detecção e correção do erro em CC1 ( $t_{APA}$ ) é igual ao atraso de propagação do caminho duplicado ( $t_{dup}$ ) menos o atraso inserido pelo multiplexador ( $t_{MUX}$ ) e pelas restrições temporais ( $t_{setup}$ ) do próximo flip-flop ( $t_{ff}$ ) e do secundário do APAFF ( $t_{ffs}$ ), como mostra a Equação 3.3.

$$t_{APA} = t_{dup} - t_{MUX} - t_{ff} - t_{ffs} \quad (3.3)$$

Quanto maior o caminho duplicado, maior a janela de detecção e correção de erros, sendo ela a soma dos atrasos de todas as portas que o compõe o caminho lógico ( $t_{porta}$ ) mais os atrasos gerados pelas trilhas e interconexões ( $t_{carga}$ ). O atraso de propagação de um caminho duplicado com  $N$  portas é mostrado pela Equação 3.4.

$$t_{dup} = \sum_{i=1}^N (t_{porta_i} + t_{carga_i}) \quad (3.4)$$

Em circuitos complexos os caminhos lógicos não são necessariamente convergentes como o diagrama da Figura 3.5 faz parecer. Vários caminhos lógicos distintos conectam entre si e levam valores para elementos sequenciais diferentes. A conexão que conduz um sinal para fora de um determinado caminho é chamada de *fan out* (WESTE; HARRIS; BANERJEE, 2005). Se um dos caminhos duplicados pelo APA propagar um sinal errôneo para fora do domínio duplicado, o sistema como um todo poderá falhar. Essas conexões são tratadas através da inserção de multiplexadores, similar ao colocado no final da duplicação. Sempre que o APA realiza a duplicação de um caminho lógico, o domínio criado é envolto por multiplexadores, que direcionam os sinais corretos para o resto do circuito.

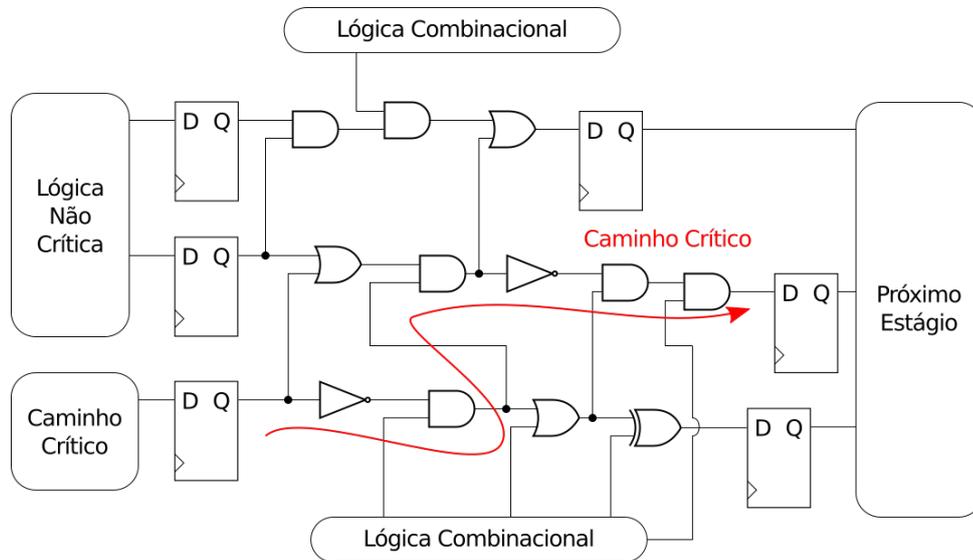
A Figura 3.6 mostra o exemplo de um circuito com o APA. O caminho crítico alvo da técnica conecta-se ao APAFF e a lógica que segue é duplicada. Os multiplexadores que realizam a interface de saída entre o domínio duplicado (representado pelas setas vermelhas) e o resto do circuito estão destacados em verde. As conexões entre os multiplexadores e o pino de erro do APAFF não foram representadas nessa imagem para facilitar o entendimento.

A temporização do circuito deve ser verificada sempre que um multiplexador é inserido. Alguns caminhos críticos podem originar a partir de outros, compartilhando as mesmas portas lógicas iniciais. A inserção de um multiplexador em meio a esses caminhos pode comprometer essa temporização e transformá-lo em uma possível fonte de erros dinâmicos. De fato, caminhos lógicos em circuitos complexos não são inteiramente únicos. Muitas estruturas são compartilhadas por diferentes partes do circuito. Um somador em uma Unidade Lógica Aritmética (ULA), por exemplo, pode receber valores de diferentes pontos do sistema. Isso resulta em caminhos lógicos que podem atravessar uns aos outros, convergir em um único ou se separar em vários.

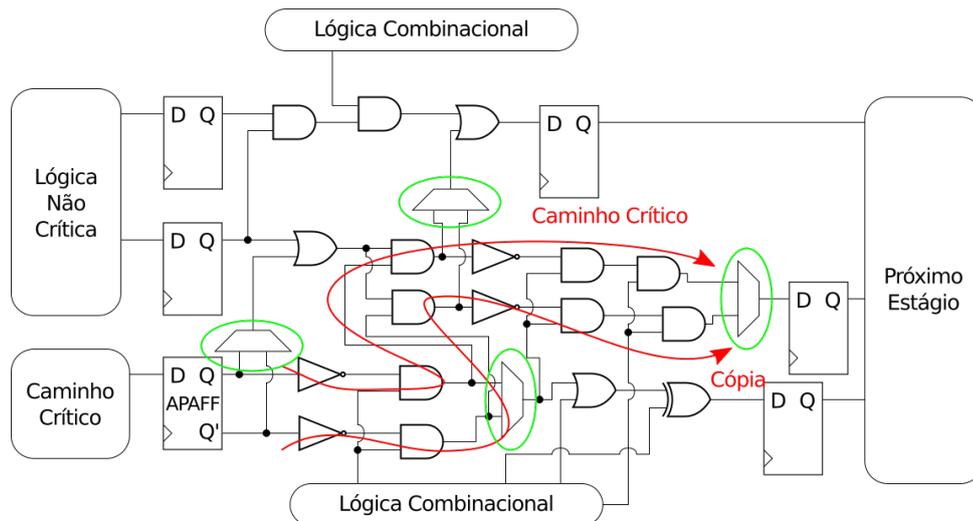
Caminhos lógicos alvos de duplicação que convergem ou atravessam outros duplicados são tratados de forma especial pelo APA. A técnica não pode invadir o domínio de um caminho previamente duplicado e copiá-lo novamente. Tal comportamento não só resultaria em um acréscimo de área como levaria a uma complexidade lógica que poderia gerar novos caminhos críticos e possíveis fontes de erros temporais. Para lidar com isso, a duplicação deve ser feita até o ponto de encontro entre o caminho em duplicação e o outro previamente duplicado.

As Figuras 3.7 mostra dois caminhos críticos, CC1 e CC3, marcados para aumento de margem temporal. O primeiro é seguido por um caminho crítico CC2 composto pelos caminhos lógicos C1 e C2, ao passo que o segundo é seguido por CC4, constituído por C1, C3 e C4. Se tanto CC2 quanto CC4 não possuem folga temporal para o emprego do

Figura 3.6 – Domínio duplicado (setas vermelhas) com envoltória de multiplexadores (verde)



(a) Caminho crítico antes da duplicação

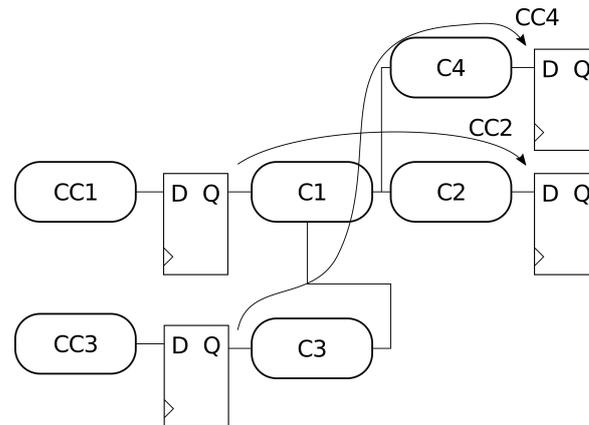


(b) Caminho após a duplicação do APA

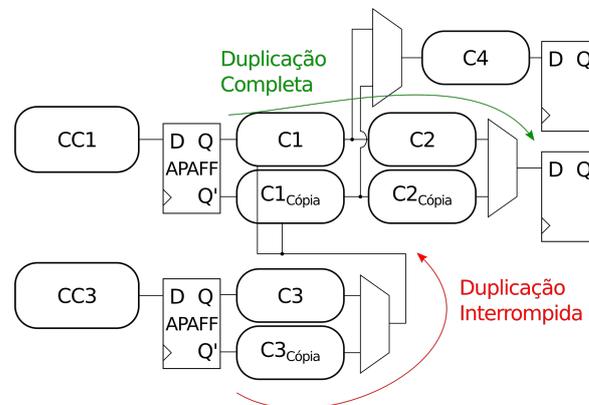
EST, então a única solução para aumentar a margem de tempo em CC1 e CC3 é através da aplicação do APA. Supondo também que CC1 é mais crítico do que CC3 e de que a inserção do APA é feita em ordem de criticidade, então poderá acontecer de CC2 ser duplicado por inteiro ao passo que CC4 não, como pode ser visto na Figura 3.7b. O trecho C1 será duplicado para atender ao APAFF de CC1 por conta da ordem de criticidade e isso gera um conflito quando CC4 é duplicado, forçando sua duplicação a ser parcial.

De fato, a duplicação total dos caminhos lógicos nem sempre é necessária e pode resultar em falsos erros. Uma mudança de valor gerada por uma lógica com baixo atraso de propagação, como mencionado na Seção 3.1, pode ser detectada como erro temporal. Essa situação é exemplificada na Figura 3.8a. Um caminho crítico e um não-crítico (CC e

Figura 3.7 – Colisão entre domínios duplicados



(a) Circuito antes da duplicação



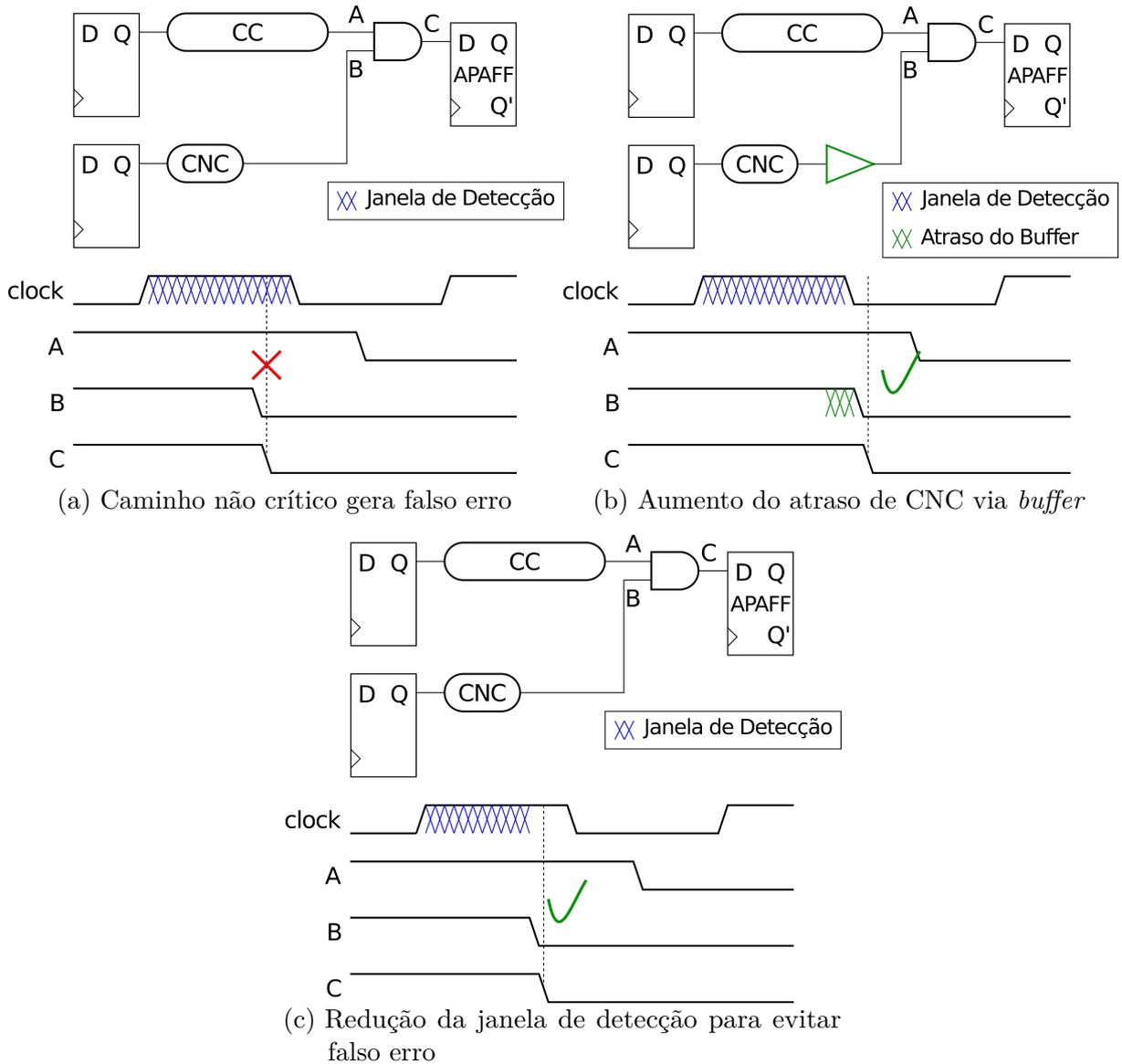
(b) Circuito após duplicação

CNC, respectivamente) conectam-se ao APAFF através de uma porta lógica AND. Se o processamento do CNC faz com que o nó “B” mude de “1” para “0” (Figura 3.8a) forçando uma transição no sinal em “C” dentro da janela de detecção, um falso erro será produzido. A transição do sinal será interpretada pelo APAFF como um erro temporal e este acionará os multiplexadores do caminho duplicado, propagando o erro para o próximo estágio. Esse problema pode ser remediado através da inserção de atraso no CNC, como mostrado na Figura 3.8b, ou através da redução do tamanho da janela de detecção como na Figura 3.8c.

A inserção de atraso é um processo caro por acrescentar área e consumo no circuito. Como visto na Seção 3.3, a forma mais simples de se atrasar um sinal é através de vários *buffers* conectados em série. Esse método também apresenta variações no atraso devido a baixa rejeição de alimentação com inversores comuns. Dessa forma, torna-se preferível reduzir a janela de detecção.

As conexões internas do circuito podem resultar em vários caminhos lógicos conectados ao domínio duplicado. Na Figura 3.9a o estágio seguinte ao CC possui dois caminhos, um composto por C1 e C2, com atraso de propagação  $t_1 + t_2$ , e outro por C1 e C3, com

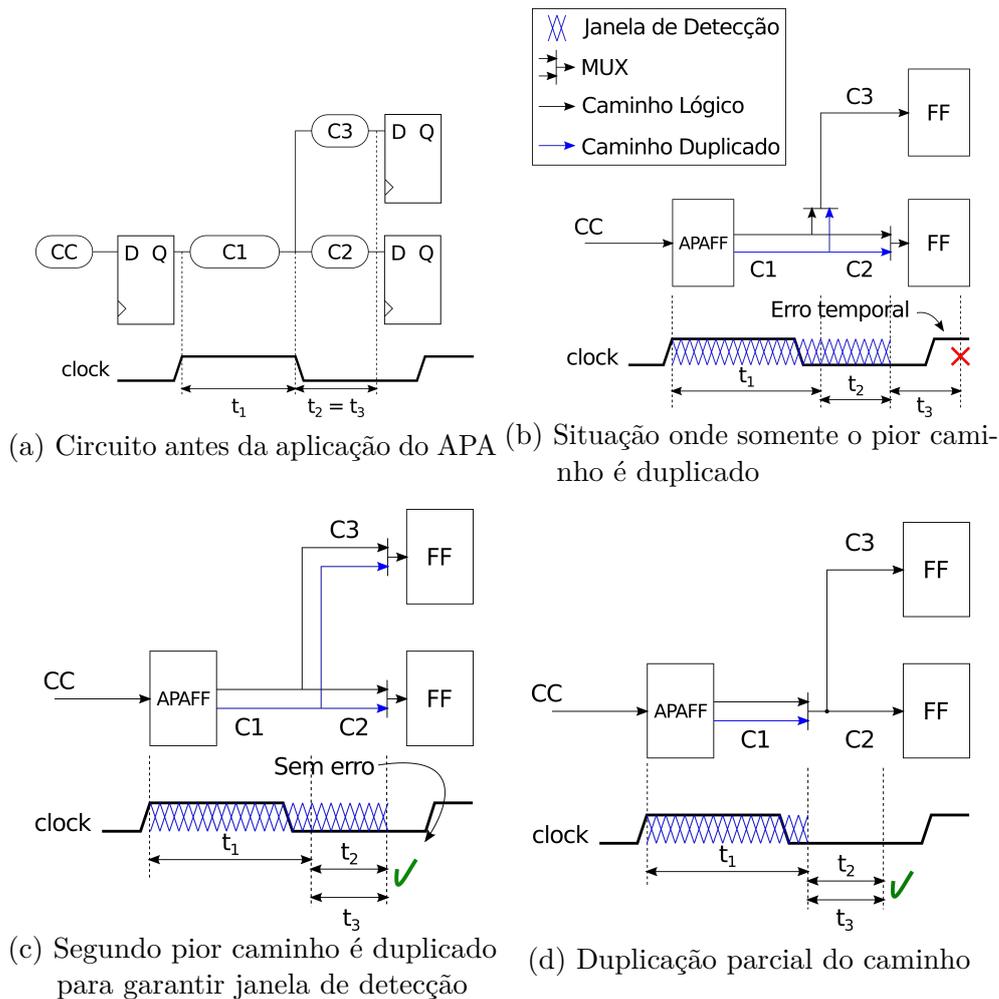
Figura 3.8 – Soluções para falso erro gerado por caminhos não críticos



atraso  $t_1 + t_3$ . Se a duplicação é feita somente em C1 e C2 de modo a resultar em uma janela de detecção proporcional a  $t_1 + t_2$ , o caminho composto por C1 e C3 pode se tornar uma fonte de erros temporais, como mostra a Figura 3.9b. Isso ocorre porque, se um erro temporal em CC acontece próximo ao final da janela de detecção, o sinal corrigido ainda levará um tempo  $t_3$  para chegar ao flip-flop conectado a C3, totalizando um atraso total de  $t_1 + t_2 + t_3$  que, nesse exemplo, é maior do que o período do *clock*. Esse problema pode ser resolvido através da duplicação dos dois caminhos, estendendo-a para os dois flip-flops como mostra a Figura 3.9c, ou através da duplicação parcial, onde a janela de detecção é reduzida pela diminuição do número de portas duplicadas como na Figura 3.9d.

Nesse exemplo, a janela de detecção possui um tamanho próximo ao do ciclo de *clock* inteiro. Como mencionado anteriormente, caminhos lógicos rápidos que estiverem conectados ao APAFF poderão ter seus sinais detectados como erros, se ocorrerem transições

Figura 3.9 – Caminho crítico originando do domínio duplicado



dentro da janela de detecção. A duplicação parcial de caminhos torna-se uma alternativa para a redução de falsos erros temporais, resultando também em menor acréscimo de área e consumo.

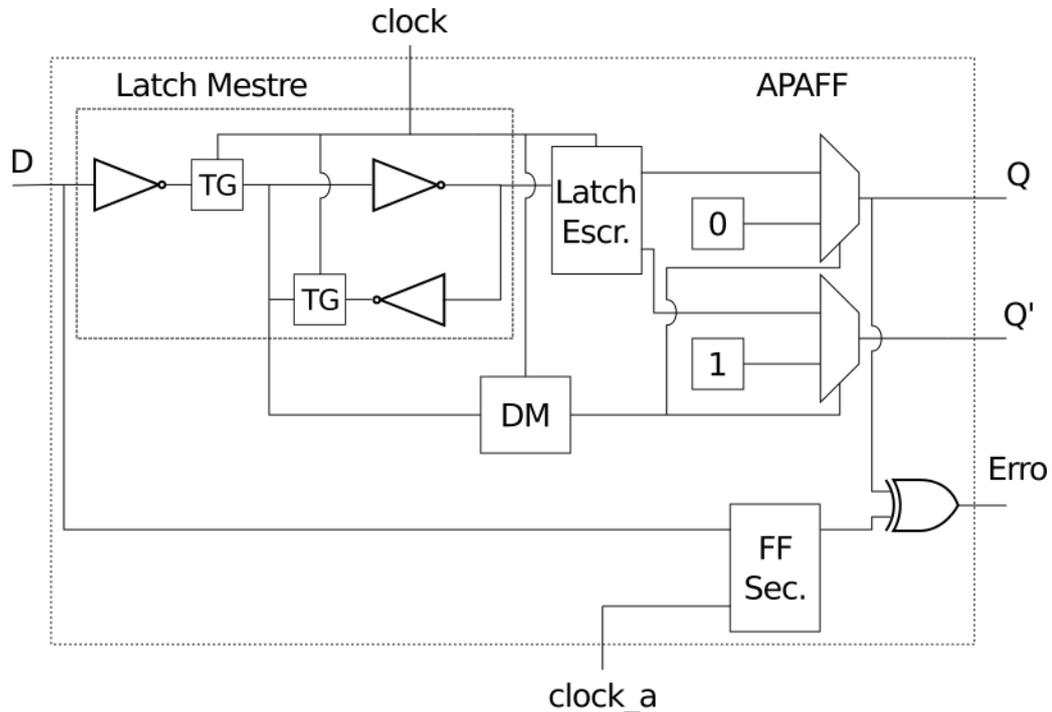
A ramificação da duplicação do APA não pode, no entanto, abranger o circuito inteiro. A colisão entre domínios duplicados impede que todo o circuito seja duplicado. O balanceamento entre quais caminhos devem ser duplicados e até onde a duplicação deve ocorrer apresenta um ponto de otimização promissor para a técnica.

### 3.4.1 Estrutura do APAFF

Ao contrário do TBFF, o APAFF utiliza um elemento sequencial secundário, para detecção do erro temporal, e um detector de metaestabilidade (DM). A Figura 3.10 ilustra essa estrutura.

Durante a operação normal o flip-flop principal, composto pelos latches mestre e escravo, amostra o sinal e passa adiante pela saída Q. O detector de metaestabilidade

Figura 3.10 – Diagrama do APAFF

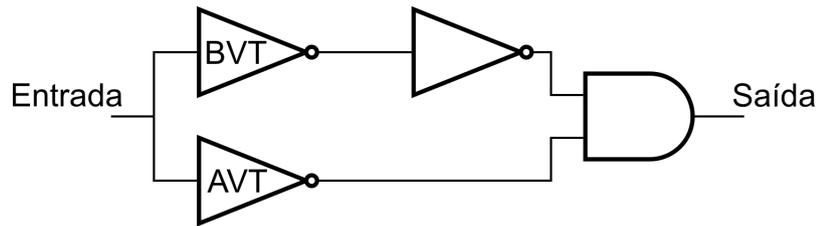


estará emitindo nível lógico baixo, comutando a saída do elemento para os pinos Q e Q'. Um flip-flop secundário (FF Sec.) realiza a amostragem do mesmo sinal na entrada D com um atraso definido pelo *clock\_a*, que estabelece a janela de detecção e correção do APA.

Se um erro temporal é detectado, o sinal *Erro* emitido pela porta lógica XOR é enviado aos multiplexadores que envolvem o domínio duplicado, comutando os sinais processados de Q para Q'. Se o flip-flop principal tornar-se metaestável, o circuito DM, conectado ao nó de realimentação do latch mestre, emitirá um sinal que chaveará os dois multiplexadores na saída do APAFF, forçando a saída Q ao nível lógico “0”, e Q’ ao “1”. Esse comportamento corta a imprevisibilidade do atraso gerado pelo flip-flop metaestável, garantindo que as saídas do APAFF decidam por um resultado estável (mesmo que errado). Os valores emitidos serão avaliados pela XOR após o atraso de detecção do flip-flop secundário, que chaveará os multiplexadores do circuito para o valor correto.

Um circuito capaz de realizar esse trabalho é o apresentado na Figura 3.11. A entrada do circuito é conectada a dois inversores, um com alta tensão de limiar (AVT) e outro com baixa (BVT). Quando o sinal de entrada encontra-se em uma tensão intermediária  $V_m$ , como é o caso do nó de realimentação de um latch metaestável, o inversor BVT processará  $V_m$  como nível lógico “1”, conseqüentemente emitindo um “0” em sua saída. O inversor normal conectado à saída de BVT inverterá o sinal novamente para “1”. Ao mesmo tempo, o inversor AVT enxergará a entrada como um nível lógico “0”, passando adiante o seu inverso “1”. Quando essa situação é cumprida, a porta E ao final do circuito emite um sinal “1”, alertando para a condição de metaestabilidade (ERNST et al., 2003).

Figura 3.11 – Detector de metaestabilidade



Fonte: Adaptado de [Ernst et al. \(2003\)](#)

O circuito do detector de metaestabilidade deve agir como uma chave de modo que, uma vez acionado, ele só voltará para o nível lógico “0” após a próxima borda de subida do *clock*. A operação do APAFF com o detector, no entanto, não é o foco deste trabalho e a sua implementação e simulação é sugerida para trabalhos futuros.

### 3.5 Fluxo de Inserção

Ambas as técnicas propostas neste documento são inseridas no circuito durante a fase de projeto. O fluxo empregado no projeto de circuitos digitais foi discutido na Seção 2.5. Essa metodologia de trabalho foi utilizada como base para formalizar a implementação do EST e APA.

Durante a síntese lógica do circuito, uma descrição em nível de registradores (*Register Transfer Logic* ou simplesmente RTL) é traduzida em lógica combinacional e sequencial. O texto de descrição comportamental do sistema - em linguagem HDL - é transformado e uma lista de conexões entre portas lógicas (*netlist*), idealmente com o mesmo comportamento que o RTL. Essa lista de conexões permite uma temporização preliminar do circuito, usando estimativas de atrasos das portas lógicas e dos fios metálicos que as conectam.

Através de ferramentas de análise estática temporal (STA) é possível descobrir os principais caminhos críticos do circuito descrito em *netlist*. Com essa informação, os caminhos lógicos conectados aos críticos podem ser avaliados para inserção do EST ou APA.

O Algoritmo 1 ilustra o procedimento de inserção das técnicas. Primeiramente, duas margens são estabelecidas pelo usuário, com valores variando de “0” à “1”, denominadas margens de caminhos críticos (*mcc*) e margem de inserção do APA (*mapa*). A *netlist* do circuito em questão é carregada a partir da ferramenta de síntese empregada junto ao fluxo. Todos os caminhos lógicos que compõem o circuito são avaliados em busca dos críticos a serem alvos das técnicas. A extração desses é feita ao captar-se todos os caminhos lógicos cujos atrasos de propagação são maiores ou iguais a  $t_{pior} * mcc$ , com  $t_{pior}$  o atraso de

propagação do caminho mais crítico do circuito. Os caminhos seguintes a cada crítico são extraídos e armazenados em um vetor ( $CS$ ). O pior atraso desses ( $t_{pdCS}$ ) é dividido por  $t_{pdCC}$  e, se o resultado estiver acima da margem estabelecida por  $mapa$ , o caminho crítico em questão será alvo do APA através da inserção do APAFF e da duplicação de todos os caminhos seguintes a ele ( $caminho_{seg}$ ). Caso contrário, o balanceamento temporal é calculado e o TBFF inserido com a quantidade adequada de *buffers* de atraso.

---

**Algoritmo 1** Inserção de APA e EST
 

---

```

1:  $t_{pior} \leftarrow$  pior atraso da netlist
2: para cada  $caminho_{crt}$  em netlist faça
3:   se atraso do  $caminho_{crt} > t_{pior} * mcc$  então
4:      $t_{pdCC} \leftarrow$  atraso do  $caminho_{crt}$ 
5:      $CS \leftarrow$  todos os caminhos seguintes ao  $caminho_{crt}$ 
6:      $t_{pdCS} \leftarrow$  pior atraso de  $CS$ 
7:     se  $t_{pdCS}/t_{pdCC} > mapa$  então
8:       Insere APAFF em  $caminho_{crt}$ 
9:       para cada  $caminho_{seg}$  em  $CS$  faça
10:        Duplica  $caminho_{seg}$ 
11:      fim para
12:     senão
13:       Calcula balanceamento temporal
14:       Insere TBFF em  $caminho_{crt}$ 
15:     fim se
16:   fim se
17: fim para
18: devolve nova netlist

```

---

As margens estabelecidas pelo usuário ajudam a estabelecer os custos envolvidos na inserção das técnicas. Quanto mais próximo de “1” for o valor de  $mcc$ , menor o número de caminhos críticos extraídos pelo fluxo, permitindo menores gastos de área e consumo, mas menor aumento de confiabilidade. O ajuste da margem  $mapa$  funciona de forma similar. Quanto maior o valor, menores caminhos serão alvos da técnica APA, com o algoritmo priorizando a aplicação do EST, uma série com menores custos, mas que oferece menor robustez.

Os *buffers* de atraso podem ser inseridos juntos aos TBFF, gerando os sinais de *clock* atrasado *in situ*, ou podem ser inseridos centralmente no circuito, estabelecendo uma árvore de distribuição de *clocks* atrasados. Nesse último caso, a inserção do TBFF consistirá somente em conectar o flip-flop crítico à árvore de atraso, resultando possivelmente em menor consumo e área para o circuito.



## 4 Simulações e Resultados

Este capítulo apresenta os resultados obtidos através de simulações funcionais e de inserção das técnicas propostas. Ao longo do texto também serão discutidos a metodologia empregada e os impactos observados em circuitos complexos.

Os resultados obtidos podem ser divididos em duas etapas. A primeira envolve a validação do EST e do APA em nível elétrico utilizando *software* de projeto e simulação de circuitos em nível de transistores. Verificado o funcionamento das técnicas, a segunda etapa consiste na inserção dessas em circuitos reais com distintos graus de complexidade para avaliação de aplicabilidade.

### 4.1 Validação das Técnicas

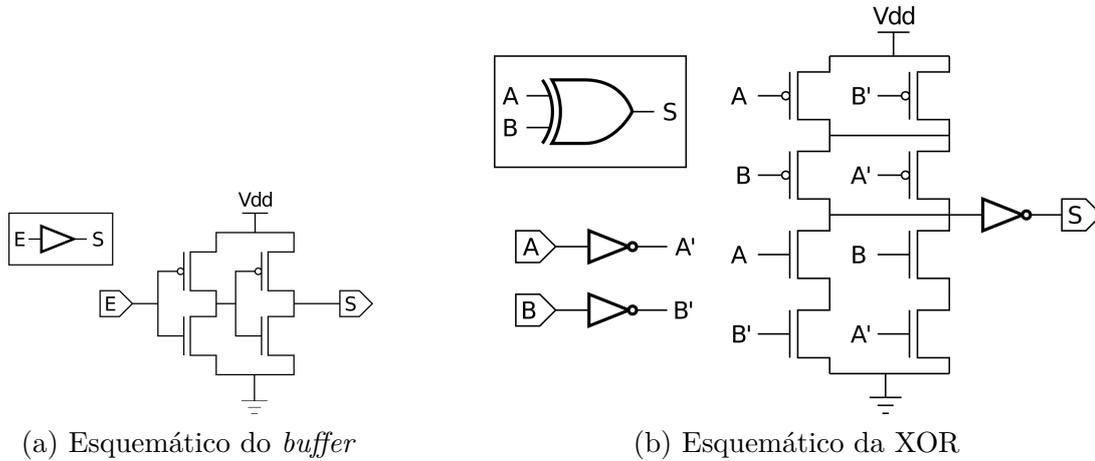
Apesar do escopo lógico-digital do trabalho, atributos como atraso e detecção de erros temporais são medidos a partir de simulações elétricas das células lógicas. Por esse motivo, a primeira etapa na avaliação das técnicas propostas envolve a simulação de seu comportamento em nível elétrico. Essas simulações se tornam necessárias para avaliar a resposta dos componentes microeletrônicos e mensurar seus efeitos nos circuitos em que se inserem.

Os *softwares* utilizados para projetar os componentes e simula-los foram o Cadence Virtuoso e o Multi-Mode Simulation with Spectre (MMSIM), respectivamente. As portas lógicas utilizam tecnologia de 130nm (nanômetros) e foram fornecidas pela *Global Foundries* através do programa MOSIS ([SERVICES, 2017](#)).

A Figura 2.8 mostra o diagrama do flip-flop empregado nas simulações. A estrutura é composta por dois latches, um mestre e um escravo, conectados em série como apresentado na Seção 2.4.1. Cada latch sincroniza-se com o *clock* através das portas de transmissão (TG1 à TG4), compostas por dois transistores nMOS e pMOS em paralelo. As Figuras 4.1a e 4.1b mostram o esquemático de um *buffer* e uma porta XOR utilizada para emitir o sinal de erro temporal, respectivamente.

O TBFF é constituído de um flip-flop tipo D comum e uma quantidade  $N_{TBFF}$  de *buffers* em sua entrada para produção do *clock* atrasado. A quantidade necessária de *buffers* é calculada a partir do atraso de propagação do caminho crítico ( $t_{pd}$ ) e o atraso inserido por um *buffer* ( $t_{buffer}$ ). O tamanho da janela de atuação do TBFF é calculado por  $t_{TBFF} = t_{pd} - t_{balanceado}$ , com  $t_{balanceado}$  já estabelecido pela Equação 3.1. Dessa forma  $N_{TBFF}$  é determinado pela Equação 4.1.

Figura 4.1 – Diagramas utilizados em simulação



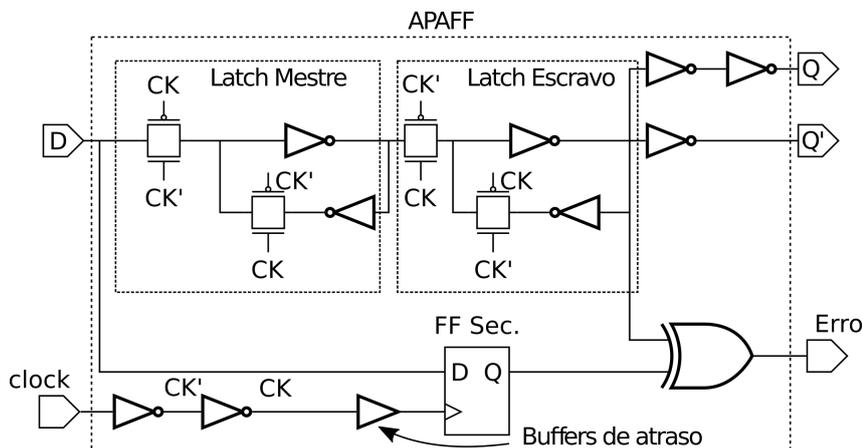
$$N_{TBFF} = \left\lceil \frac{t_{TBFF}}{t_{buffer}} \right\rceil \quad (4.1)$$

No caso do APA, a janela de detecção e correção é calculada a partir do caminho duplicado do estágio seguinte ao crítico, como mostrado na Equação 3.3. Da mesma forma que no TBFF, a quantidade de *buffers* necessários para produzir a janela do APA ( $N_{APA}$ ) é calculado pela Equação 4.2.

$$N_{APA} = \left\lceil \frac{t_{APA}}{t_{buffer}} \right\rceil \quad (4.2)$$

A estrutura do APAFF utilizada na simulação é mostrada na Figura 4.2. O detector de metaestabilidade não foi simulado durante a validação das técnicas a fim de simplificar o processo e focar no funcionamento do APA em si. Uma análise da metaestabilidade observada durante essas simulações é conduzida na Seção 4.1.4.

Figura 4.2 – APAFF simulado no software MMSIM

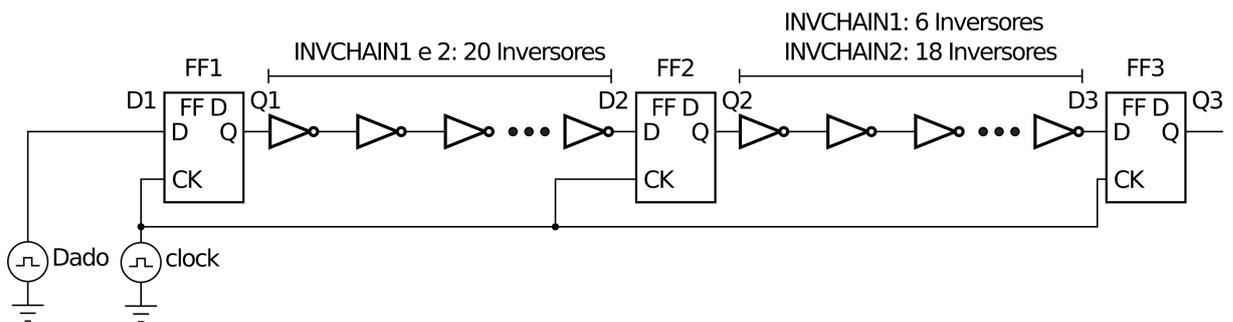


### 4.1.1 Circuitos de Validação

Três circuitos foram utilizados para validar o funcionamento do EST e APA. Os dois primeiros são chamados INVCHAIN1 e INVCHAIN2 e são compostos por dois estágios sequenciais conectados entre si por cadeias de inversores. O terceiro, denominado COMPRESSOR, é composto por três somadores também separados em dois estágios com uma configuração do tipo compressor 4:2 de 8-bits.

Os circuitos INVCHAIN utilizam uma cadeia de 20 inversores conectados em série no primeiro estágio. O segundo estágio é composto por 6 inversores no INVCHAIN1 e 18 no INVCHAIN2. O primeiro circuito possibilita observar a atuação do EST devido a presença de folga temporal, ao passo que o segundo representa um circuito onde essa folga não está presente e, portanto, requer a aplicação do APA.

Figura 4.3 – Diagrama do circuito INVCHAIN com diferentes valores de inversores para os dois circuitos



Nas simulações realizadas com o EST, os circuitos são modificados com *buffers* de atraso, representados pelo bloco  $\Delta$ , inseridos no pino de *clock* (CK) do flip-flop FF2. Para a validação do APA, a cadeia de 18 inversores seguinte ao crítico foi duplicada além da substituição de FF2 por um APAFF. Um multiplexador ao final do caminho duplicado é controlado pelo sinal de erro (pino E) do APAFF. Essas mudanças são mostradas na Figura 4.4.

O circuito compressor 4:2, como mencionado anteriormente, é composto por dois somadores que realizam a operação em quatro valores de entrada. A Figura 4.5 mostra um diagrama simplificado de um compressor. Esse circuito é utilizado em multiplicadores onde muitas operações de soma devem ser realizadas simultaneamente para obter um melhor desempenho (BECHADE, 1998). O objetivo de seu emprego na validação das técnicas foi de mostrar o funcionamento destas em circuitos mais elaborados, onde estão presentes vários caminhos críticos.

O COMPRESSOR é uma versão alterada do circuito encontrado em literatura. De forma a manter os caminhos críticos evidentes, três somadores do tipo *Ripple-carry* foram utilizados, composto por módulos chamados somadores completos (*Full-adders*) e por um meio somador (*half adder*) como mostra a Figura 4.6. Um módulo de somador completo



Figura 4.6 – Somador do tipo *ripple-carry*

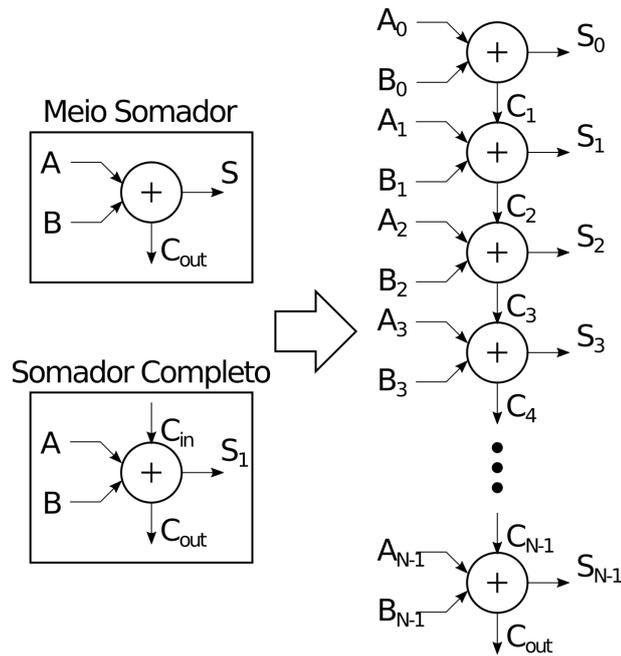
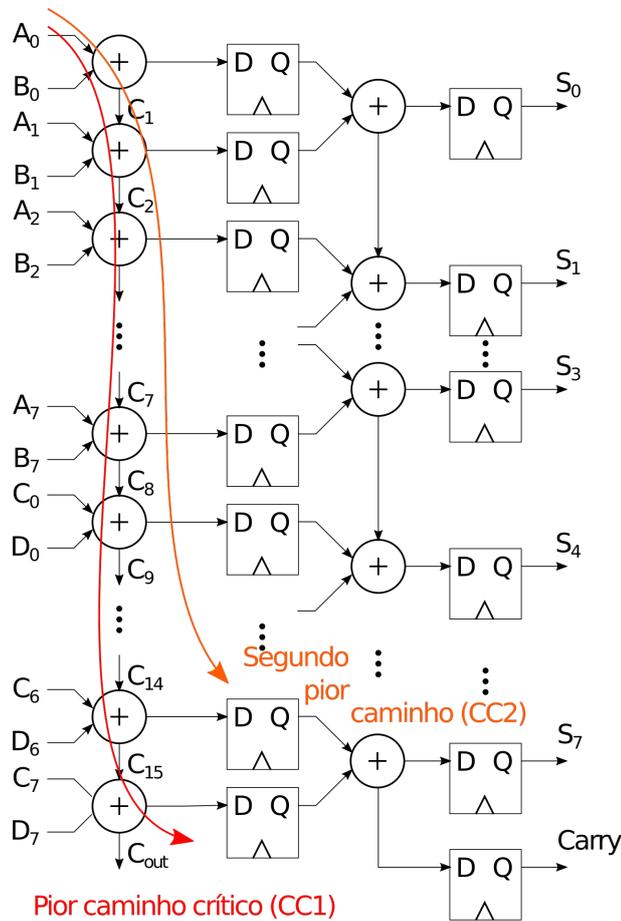


Figura 4.7 – Circuito COMPRESSOR utilizado em simulação



de simulação não é de fácil automação para essas tarefas.

### 4.1.2 Metodologia de Simulação

Apesar das técnicas visarem o aumento da vida útil dos dispositivos em que se inserem, os testes foram realizados não com a degradação das células que compõem os circuitos, mas através do aumento da frequência de operação. A diminuição do período de *clock* que sincroniza o circuito reduz o tempo que a lógica combinacional possui para o processamento dos sinais, como visto na Seção 2.4. Esse método, apesar de simplista ao se testar uma técnica voltada para mitigar problemas de degradação, apresenta uma abordagem rápida e prática para se observar as mudanças no circuito.

O fluxo de simulação adotado para os circuitos de teste são compostos por três etapas. A primeira é o cálculo do balanceamento temporal no caso do EST, ou da janela de detecção e correção do APA, como já estabelecido pelas Equações 3.1 e 3.3. Esses cálculos são feitos através do atraso de propagação da lógica combinacional, o atraso dos multiplexadores e os tempos  $t_{setup}$  dos flip-flops, todos obtidos através de simulações. A etapa seguinte consiste no cálculo da quantidade de *buffers* de atraso necessários para se alcançar os valores calculados anteriormente. As simulações são feitas em seguida com diferentes quantidades de *buffers* e medindo-se as frequências máximas dos circuitos modificados.

A frequência máxima é extraída dos circuitos através de um fluxo automático escrito em linguagem SKILL, utilizada na programação do *software* de simulação (BRUNVAND, 2010). O fluxo é ilustrado pelo Algoritmo 2. A frequência máxima de operação é encontrada através de um método numérico denominado Método da Bissecção (FAIRES; BURDEN, 1998). Nele, o usuário estabelece uma janela definida pelo valor máximo e mínimo de frequência ( $f_{max}$  e  $f_{min}$ , respectivamente). O valor de frequência a ser testado ( $f_{teste}$ ) é calculado em seguida a partir da média entre  $f_{min}$  e  $f_{max}$ . A simulação é executada com frequência  $f_{teste}$  e o resultado é avaliado. Se o circuito retornou um valor correto, então a frequência pode ser aumentada, com  $f_{min}$  recebendo o valor de  $f_{teste}$ . Caso contrário, o circuito deixou de funcionar e a frequência é reduzida, então  $f_{max}$  recebe o valor de  $f_{teste}$ . O valor de  $f_{teste}$  é atualizado em seguida e uma nova iteração começa. A janela  $f_{max} - f_{min}$  encolhe até uma margem estabelecida pelo usuário, quando o programa encerra sua operação e retorna a frequência máxima encontrada.

Esse fluxo foi executado para diferentes números de *buffers* no EST e APA, obtendo-se as frequências máximas para os diferentes empréstimos temporais e janelas de detecção de erro.

### 4.1.3 Resultados de Simulação

Os circuitos de teste foram simulados no *software* Multi-Mode Simulation with Spectre (MMSIM) 10.1. Nos circuitos INVCHAIN três ciclos de *clock* são necessários para

**Algoritmo 2** Frequência máxima do circuito

---

```

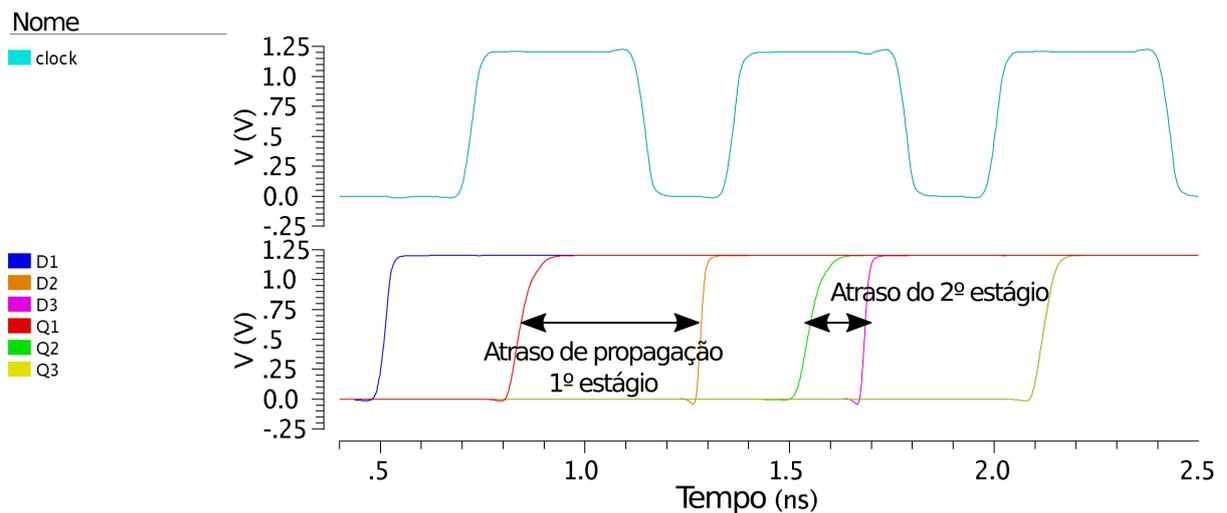
1:  $f_{teste} \leftarrow (f_{max} + f_{min})/2$ 
2: enquanto  $f_{max} - f_{min} > margem$  faça
3:    $resultado \leftarrow \text{simulação}(f_{teste})$  {Simula circuito com frequência  $f_{teste}$ }
4:   se  $resultado$  está correto então
5:      $f_{min} \leftarrow f_{teste}$ 
6:   senão
7:      $f_{max} \leftarrow f_{teste}$ 
8:   fim se
9:    $f_{teste} \leftarrow (f_{max} + f_{min})/2$  {Atualiza  $f_{teste}$ }
10: fim enquanto
11: imprime  $f_{teste}$ 

```

---

passar um valor ao longo dos dois estágios do circuito, um para inserir o valor na cadeia e outros dois para que os resultados sejam passados adiante. Na Figura 4.8 observa-se as respostas do circuito sem técnica aplicada a ele. Os sinais possuem a nomenclatura definida pelo diagrama da Figura 4.3. O atraso entre “Q1” e “D2” é o tempo que o sinal inserido no primeiro estágio leva para propagar até a entrada D do flip-flop de interface com o segundo. Da mesma forma, o tempo entre “Q2” e “D3” estabelece o atraso de propagação para o segundo estágio do circuito. O sinal “Q3” é a resposta final do circuito, amostrado por FF3.

Figura 4.8 – Resposta do circuito INVCHAIN1 simulado



Os atrasos de propagação medidos para o primeiro e segundo estágio de inversores é de 442,1 ps e 135,5 ps ( $t_{pd1}$  e  $t_{pd2}$ ), respectivamente. Cada *buffer* utilizado é composto por dois inversores de atraso 22,2 ps cada. Pela Equação 3.1 e 4.1:

$$t_{balanceado} = \frac{(442,1 \text{ ps} + 135,5 \text{ ps})}{2} = 288,8 \text{ ps} \quad (4.3)$$

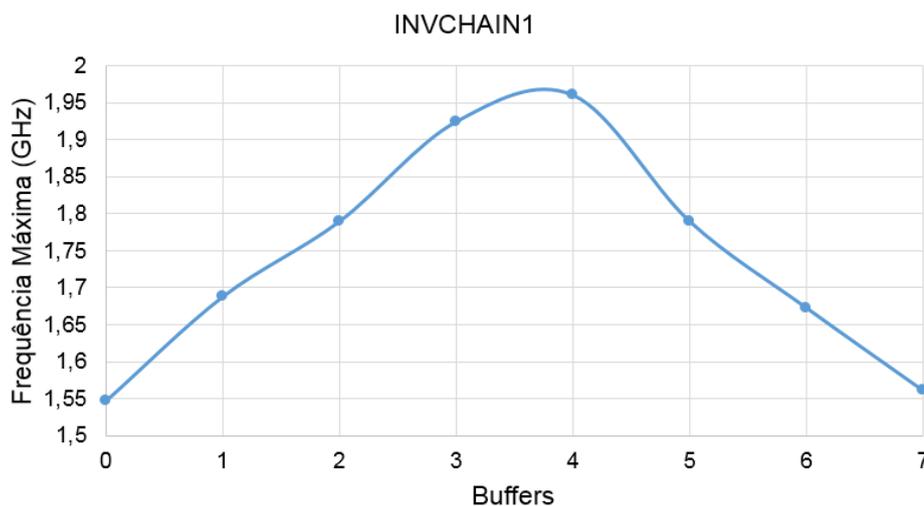
$$N_{TBFF} = \left\lceil \frac{(442,1 \text{ ps} - 288,8 \text{ ps})}{44,4 \frac{\text{ps}}{\text{buffer}}} \right\rceil = 4 \text{ buffers} \quad (4.4)$$

Espera-se que, inserindo a quantidade de *buffers* calculada na Equação 4.4, o circuito possuirá uma maior frequência de operação do que sem a técnica. O número calculado também corrobora com o emprego da técnica de empréstimo temporal, já que o estágio seguinte ao crítico possui folga suficiente para se realizar o balanceamento. Os resultados de frequência obtidos com diferentes quantidades de *buffers* pode ser visto na Tabela 4. Um gráfico com os mesmos valores também é apresentado na Figura 4.9.

Tabela 4 – Resultados obtidos aplicando-se o EST ao INVCHAIN1

Buffers	Período Crítico (ps)	Frequência Máxima (GHz)
0	656,5	1,547
1	592,3	1,688
2	558,6	1,790
3	518,5	1,925
4	509,8	1,961
5	558,6	1,790
6	597,7	1,673
7	640,5	1,561

Figura 4.9 – Gráfico com resultados da tabela 4



Pelo gráfico da Figura 4.9 é possível observar que a frequência máxima de operação do circuito é realmente maior nas imediações do número calculado de *buffers* na Equação 4.4. A medida que se aumenta o atraso no TBFF a lógica composta por 6 inversores passa a ser mais crítica do que a composta por 20, o que resulta na redução da frequência máxima do circuito.

Os acréscimos de área e consumo foram calculados para a quantidade de *buffers* que obteve maior frequência máxima. O circuito INVCHAIN com 4 *buffers* de atraso é 15,8 % maior do que o original e consome 12,0 % a mais em média.

O mesmo comportamento pode ser visto no circuito INVCHAIN2. Nesse, o circuito é composto por 20 inversores seguido por 18. A quantidade necessária de *buffers* é calculada da mesma forma para o EST. Dessa vez, o atraso  $t_{pd2}$  é de 398, 2 ps.

$$t_{balanceado} = \frac{(442, 1 ps + 398, 2 ps)}{2} = 420, 2 ps \quad (4.5)$$

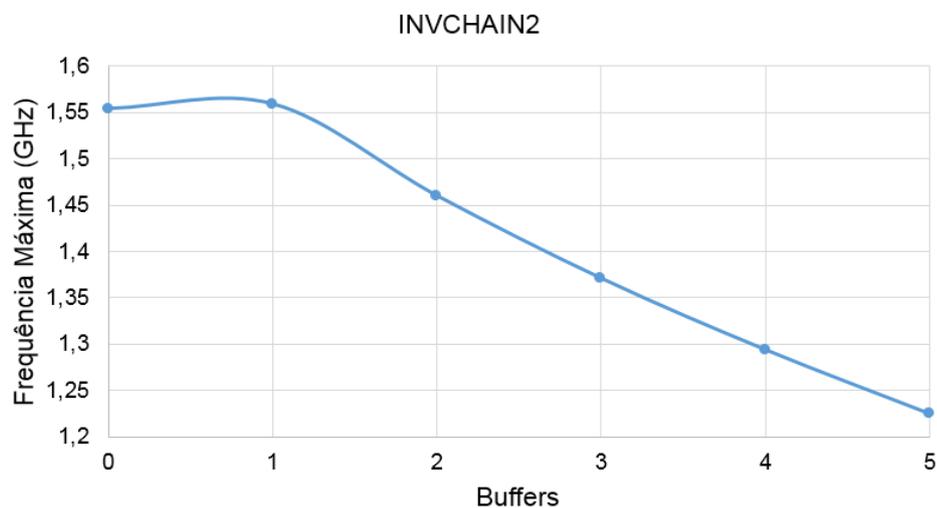
$$N_{TBF} = \left\lceil \frac{(442, 1 ps - 420, 2 ps)}{44, 4 \frac{ps}{buffer}} \right\rceil = 1 buffers \quad (4.6)$$

A quantidade de *buffers* para se obter o melhor balanceamento temporal cai para um, o que indica a baixa efetividade do empréstimo temporal neste circuito. Variando-se o número de *buffers* no flip-flop entre estágios do INVCHAIN2 obtém-se os resultados mostrados na Tabela 5 e Figura 4.10.

Tabela 5 – Resultados obtidos do EST no INVCHAIN2

Buffers	Período Crítico (ps)	Frequência Máxima (GHz)
0	643,6	1,554
1	641,5	1,559
2	685,3	1,460
3	729,0	1,371
4	772,8	1,294
5	816,5	1,225

Figura 4.10 – Gráfico com dados da Tabela 5



Dado o mal desempenho do EST com o INVCHAIN2, o APA é aplicado. O estágio seguinte ao crítico é duplicado para se obter o processamento paralelo dos dois possíveis resultados do flip-flop. Para os cálculos da janela de detecção e correção de erro e a quantidade de *buffers* de atraso são utilizadas as Equações 3.3 e 4.2. O multiplexador utilizado possui um atraso de 48,0 ps e o valor de  $t_{setup}$  é de 92,7 ps, para os flip-flops do segundo estágio e secundário. Os valores obtidos são:

$$t_{APA} = 398,2 ps - 48,0 ps - 92,7 ps - 92,7 ps = 164,8 ps \quad (4.7)$$

$$N_{APA} = \left\lceil \frac{164,8 ps}{44,4 ps} \right\rceil = 4 buffers \quad (4.8)$$

Os resultados obtidos para essas simulações são mostrados na Tabela 6 e Figura 4.11. Como calculado, a frequência máxima do circuito é maior com 3 *buffers* de atraso no APAFF. O ganho de frequência do circuito aproxima-se ao obtido no EST para o INVCHAIN1, com a importante diferença de que, nesse caso, nenhum empréstimo temporal está sendo realizado. O ganho observado ocorre devido ao processamento paralelo do segundo estágio duplicado.

Tabela 6 – Resultados obtidos de simulação com APA aplicado ao INVCHAIN2

Buffers	Período Crítico (ps)	Frequência Máxima (GHz)
0	643,6	1,554
1	591,6	1,690
2	551,3	1,814
3	510,9	1,957
4	528,2	1,893
5	574,3	1,741

O aumento de área e consumo no circuito com maior frequência máxima refletem o impacto que o APA possui no circuito. O INVCHAIN2 com duplicação e 3 *buffers* de atraso é 54,0% maior do que o circuito original e consome 74,8% mais.

O gráfico da Figura 4.12 mostra as respostas obtidas na simulação com 3 *buffers* de atraso. O sinal “Erro” indica que o resultado do caminho crítico chegou dentro da janela de detecção e correção do APA, ativando o caminho alternativo para o flip-flop FF3.

As melhoras observadas nos dois circuitos INVCHAIN com as técnicas aplicadas são casos otimistas. As cadeias de inversores não refletem a complexidade de circuitos reais encontrados na prática. Para melhor generalizar a aplicação das técnicas foi utilizado o circuito COMPRESSOR que possui maior complexidade de interconexões e apresenta vários caminhos críticos direcionados a diferentes flip-flops.

Figura 4.11 – Gráfico com resultados da Tabela 6

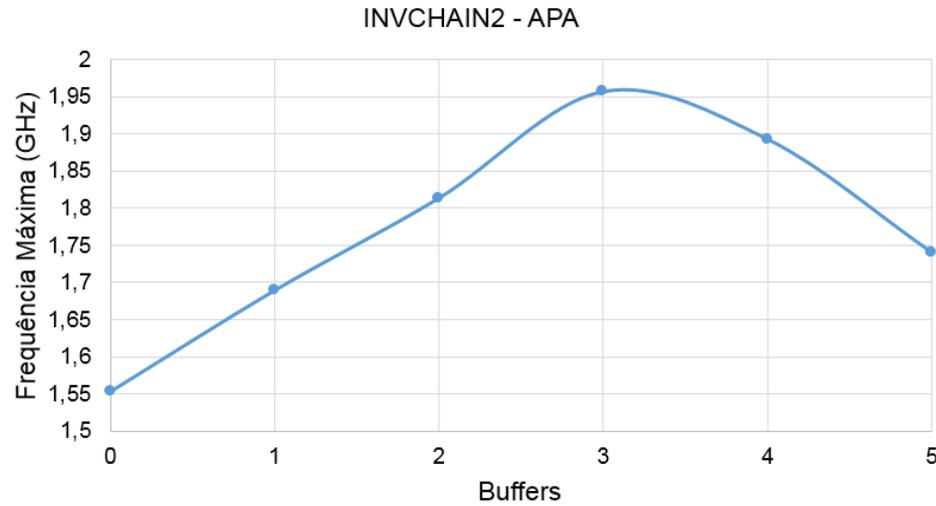
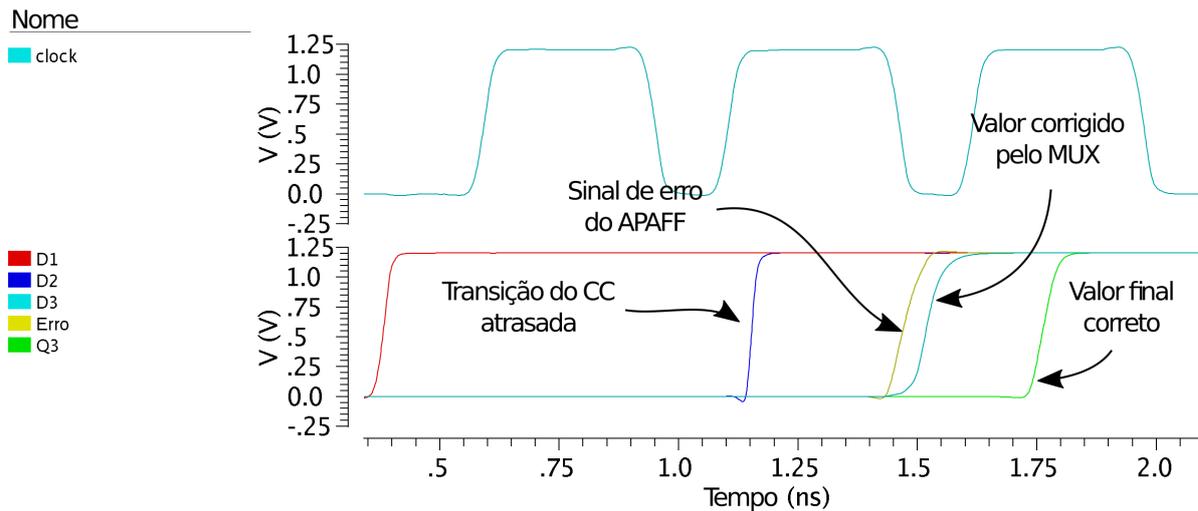


Figura 4.12 – Resposta do circuito INVCHAIN2 com APA



O pior e o segundo pior caminhos críticos (CC1 e CC2) presentes no primeiro estágio do circuito foram simulados com atrasos de 1,705 ns e 1,593 ns, respectivamente. A lógica do segundo estágio conectado ao flip-flop de CC1 é composta por um único somador completo, como mostrado na Figura 4.7. O atraso de propagação desse único somador foi simulado em 0,133 ns. Aplicando-se o EST no COMPRESSOR considerando-se somente CC1, obtém-se os seguintes resultados:

$$t_{balanceado} = \frac{1,705 \text{ ns} + 0,133 \text{ ns}}{2} = 0,920 \text{ ns} \quad (4.9)$$

$$N_{TBFF} = \left\lceil \frac{1,705 \text{ ns} - 0,920 \text{ ns}}{44,4 \text{ ps}} \right\rceil = 18 \text{ buffers} \quad (4.10)$$

Os resultados das simulações são mostrados na Tabela 7. Os valores obtidos mostram

que, apesar de CC1 continuar emitindo resultados corretos em torno de 555MHz, o circuito como um todo emite um resultado errado por causa de CC2. O empréstimo temporal conduzido em CC1 o protege do erro dinâmico, mas não a CC2. Esses resultados ilustram a complexidade da aplicação das técnicas em circuitos reais. Aplicando-se o APA ao invés do EST surte o mesmo resultado, já que o problema não está em CC1, mas na presença de outro caminho crítico com temporização próxima a dele.

Tabela 7 – Resultados das simulações com EST aplicado ao CC1 do COMPRESSOR

Buffers	Período Crítico (ns)	Frequência Máxima (MHz)
0	1,934	517,2
1	1,893	528,3
2	1,836	544,7
3*	1,797	556,5
4*	1,748	572,1

\* Erro temporal em CC2

Os valores de frequência máxima obtidos ao simular o circuito com o APA são numericamente iguais aos da Tabela 7. O aumento de área e consumo para o COMPRESSOR com EST é em torno de 4,7% e 3,9% a mais do que o original, respectivamente. No circuito com o APA, esses valores são de 5,9% e 6,2% para área e consumo.

Alterando-se o cálculo do empréstimo temporal e da janela de detecção e correção tem-se o número aproximado de *buffers* para realizar o balanceamento do pior caminho crítico. Nesse caso, o atraso de propagação do segundo pior caminho ( $t_{pdCC2}$ ) tem que ser utilizado para determinar o tamanho máximo da janela de empréstimo temporal ou detecção de erro. Esse valor deve ser subtraído do atraso do pior caminho ( $t_{pdCC1}$ ) e dividido pelo atraso do buffer individual ( $t_{buffer}$ ). A fórmula junto ao cálculo realizado para o caso do COMPRESSOR é mostrado na Equação 4.11.

$$N_{TBFF} = \left\lceil \frac{t_{pdCC1} - t_{pdCC2}}{t_{buffer}} \right\rceil = \left\lceil \frac{1,705 \text{ ns} - 1,593 \text{ ns}}{44,4 \text{ ps}} \right\rceil = 3 \text{ buffers} \quad (4.11)$$

Como visto nos valores da Tabela 7, em torno de 3 *buffers* o resultado do cômputo passa a ser errado, sendo necessário o arredondamento da quantidade de *buffers* para baixo. Evita-se assim erros temporais em outros caminhos críticos do circuito.

Circuitos complexos exigem que a técnica seja aplicada ao longo de uma faixa de caminhos críticos para que o ganho seja significativo. Os resultados obtidos nos dois circuitos do INVCHAIN mostram que é possível um ganho considerável de confiabilidade no que diz respeito a erros temporais em caminhos críticos. No primeiro circuito, o ganho de frequência observado foi em torno de 24,4% com o EST. O segundo apresentou um

problema para o empréstimo temporal devido a baixa disponibilidade de folga para ser realocada ao caminho crítico. Nesse caso, o APA obteve um ganho de quase 26 % em frequência com o número ótimo de *buffers*.

Com o circuito de teste COMPRESSOR, o ganho de frequência máxima foi limitado pelo segundo caminho crítico do circuito. Isso gerou um caso em que a frequência máxima poderia ser aumentada por não haver erros no caminho abarcado pelo EST/APA, mas outros caminhos se tornaram geradores de erros temporais, ocasionando em resultados errados do circuito como um todo.

Os impactos negativos de área e consumo das técnicas foram calculados para os casos em que a maior frequência máxima foi observada e os valores diferem consideravelmente para os diferentes circuitos de teste. O APA gerou um aumento de área de 54,0 % no INVCHAIN2 ao passo que somente 5,9 % no COMPRESSOR, por exemplo.

Acréscimos de área e consumo são influenciados pela complexidade do circuito em questão. O INVCHAIN é composto por duas cadeias de inversores com interconexões simples. A duplicação do caminho seguinte ao primeiro estágio, por exemplo, gera um grande impacto no circuito. A maior complexidade do COMPRESSOR, por outro lado, mostra que as técnicas possuem impactos muito mais singelos, pelo menos quando direcionadas somente ao pior caminho crítico do circuito.

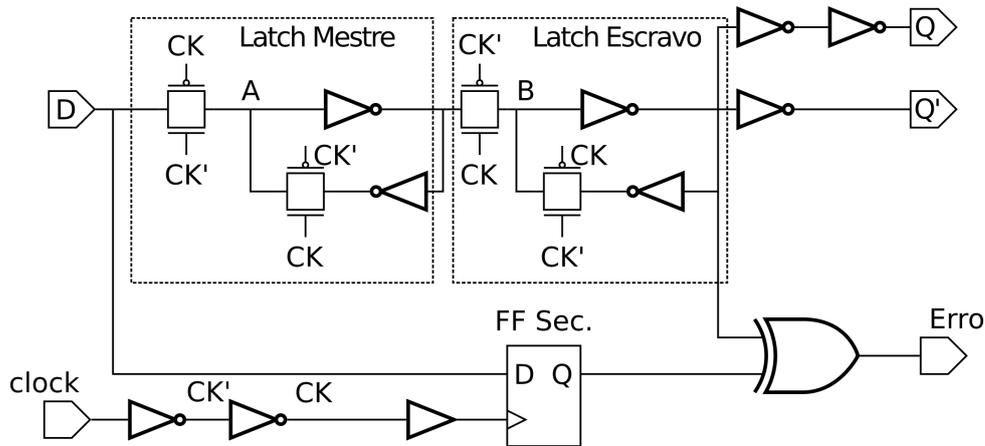
#### 4.1.4 Metaestabilidade na Validação

A metaestabilidade foi apresentada na Seção 2 e seu maior problema consiste na incerteza do atraso que o elemento sequencial metaestável insere em sua resposta. Muitos trabalhos foram desenvolvidos direcionados para a detecção e correção de erros temporais tentando abarcar a metaestabilidade e proteger o circuito desses efeitos indesejáveis, como em [Cannizzaro et al. \(2015\)](#) e [Tadros et al. \(2016\)](#).

Durante as simulações, a metaestabilidade foi observada em diversas situações mas somente apresentou problemas em algumas condições específicas. O EST possui maior resiliência à esse efeito por não lidar com amostragens adicionais do valor de entrada, com o TBFF só chegando a metaestabilidade quando o sinal muda próximo à borda de subida do *clock* atrasado. Como visto anteriormente, nessa condição o circuito já está chegando ao final da vida útil estendida provida pela técnica. O APA, por outro lado, utiliza duas bordas de amostragem, culminando em dois períodos vulneráveis à metaestabilidade, um nas proximidades da borda de subida do *clock* comum e o outro próximo à borda de subida do *clock* atrasado. Esse último período, assim como no caso do EST, representa um problema para os circuitos que já tiveram a sobrevida esperada das técnicas.

A Figura 4.13 mostra o esquemático do APAFF simulado ressaltando os nós de realimentação A e B importantes para análise da metaestabilidade.

Figura 4.13 – Esquemático do flip-flop central simulado

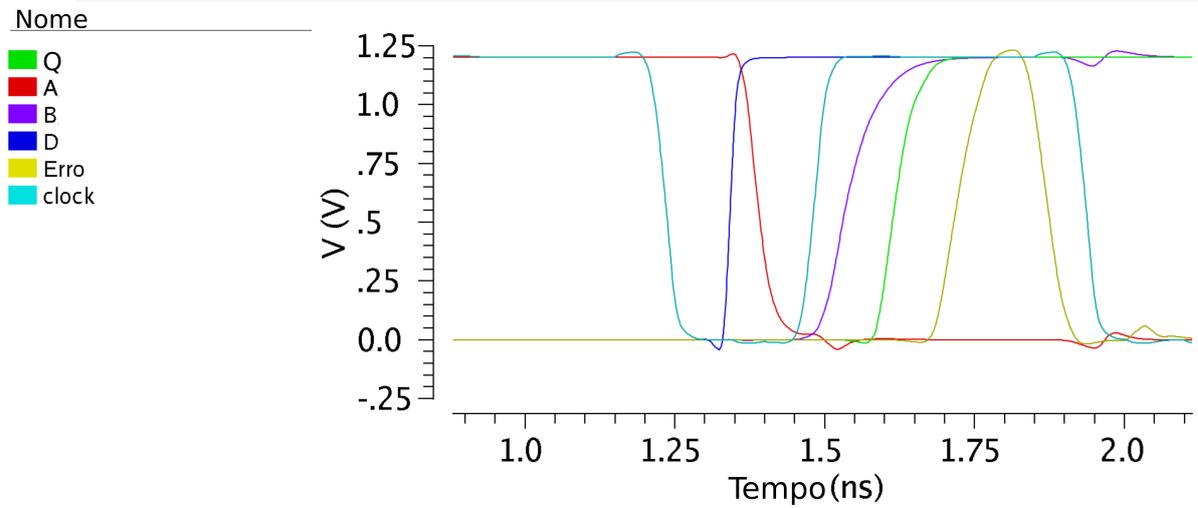


Os gráficos da Figura 4.14 mostram o fenômeno como foi observado durante as simulações no MMSIM. Na Figura 4.14a o APAFF está em operação normal, com o sinal de entrada estabilizando com antecedência antes da borda de subida do *clock*. Os nós A e B apresentam transições monotônicas de tensão e o atraso de propagação da borda de subida do *clock* para a saída Q ( $t_{pcq}$ ) possui um atraso em torno de 150 ps. Na Figura 4.14b a diminuição do período de *clock* faz com que a transição do sinal em D torne o latch mestre metaestável. Esse fenômeno pode ser observado pela tensão A que não mais decresce constantemente, flutuando em uma tensão intermediária de nível lógico indefinido. Essa indeterminação se propaga para o latch escravo através da tensão do nó B que, assim como A, oscila em uma nível intermediário antes de alcançar a estabilidade. Nessa situação, o atraso  $t_{pcq}$  é próximo a 300 ps, o dobro de tempo em relação ao funcionamento normal, e pode resultar em falhas temporais no estágio seguinte ao crítico. A Figura 4.14c mostra uma situação em que o período do *clock* foi reduzido ainda mais. Nesse caso, a metaestabilidade não se propaga para o latch escravo, impedindo a emissão de uma saída com atraso elevado.

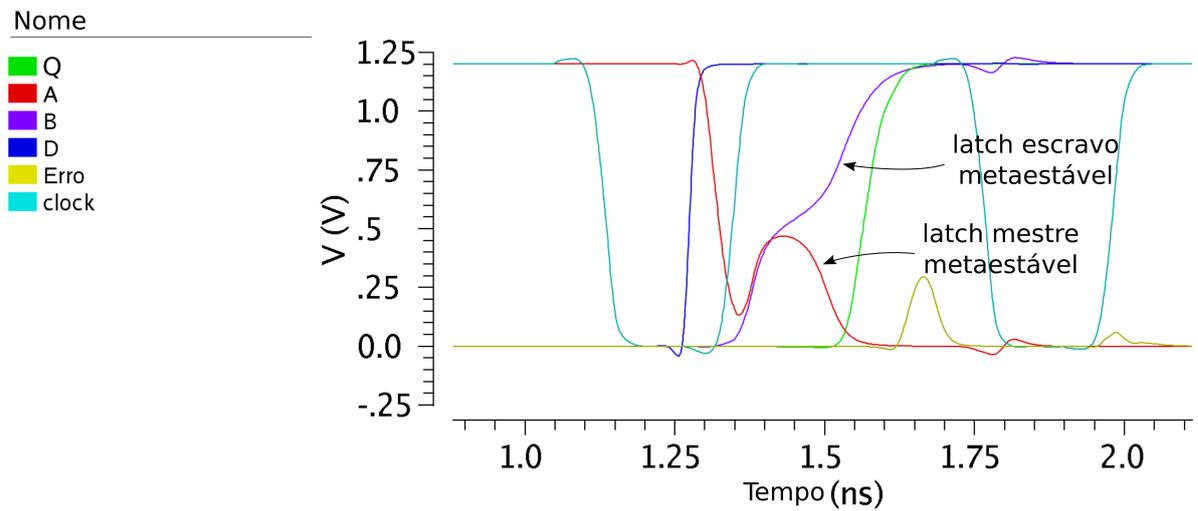
A janela de tempo em que a metaestabilidade se propaga do latch mestre para escravo no flip-flop principal é o único instante em que esse efeito prejudica a operação do APAFF. Nas simulações realizadas com tecnologia de 130 nm, essa janela possui duração entre 20 a 30 ps.

O emprego de um detector pode resolver esse problema e eliminar a janela de metaestabilidade do APAFF. No entanto, nenhum dispositivo com essa finalidade foi testado durante as simulações e o seu emprego é proposto como trabalho futuro no aprimoramento das técnicas.

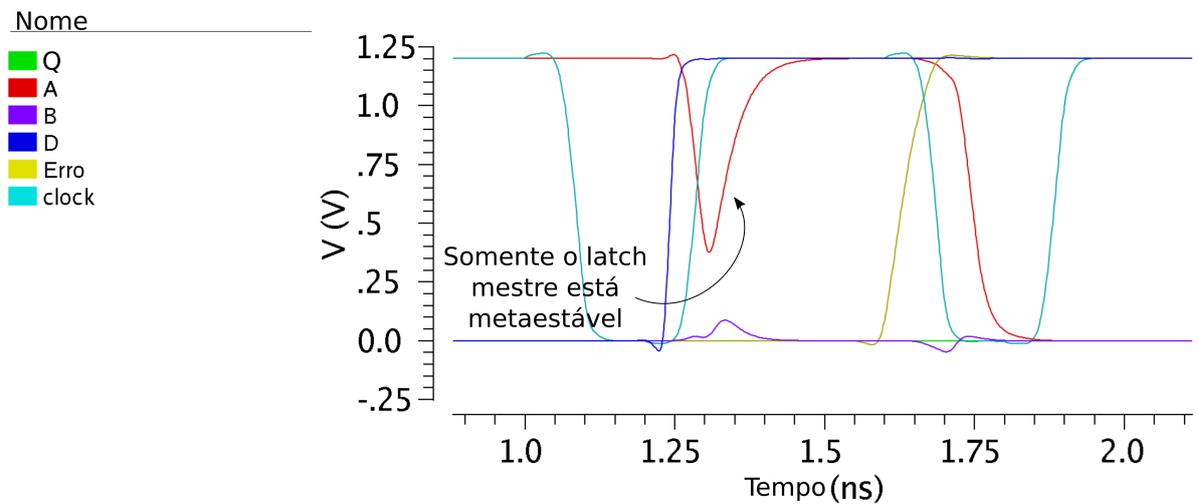
Figura 4.14 – Situações de metaestabilidade observadas no APAFF



(a) APAFF sem metaestabilidade



(b) Metaestabilidade se propagando do latch mestre para escravo



(c) Metaestabilidade não se propaga para o latch escravo

## 4.2 Inserção em Circuitos Complexos

Os resultados obtidos na etapa de validação mostram o bom desempenho das técnicas em circuitos simples, garantindo ganhos significativos de frequência máxima. No entanto, a implantação do EST e APA foi realizada manualmente nesses circuitos, prática inviável em sistemas mais complexos. Mesmo circuitos de aplicações simples podem chegar a dezenas de milhares de portas lógicas e centenas de estágios sequenciais, inviabilizando a inserção manual de *buffers* de atraso ou duplicação de caminhos.

Com esses casos em mente, um fluxo automático foi desenvolvido em linguagem de programação Tcl para o *software* de síntese lógica Cadence RTL Compiler versão 14.2. Esse fluxo realiza a inserção automática do EST e APA, demandando o mínimo de intervenção do usuário para tal. Diversos circuitos reais foram utilizados para o teste de inserção e os resultados obtidos são discutidos ao longo desta seção.

### 4.2.1 Métricas de Inserção

As técnicas propostas afetam caminhos críticos individuais, como visto nas seções anteriores. Circuitos reais requerem que ambas sejam aplicadas diversas vezes em seus caminhos para obter o aumento de robustez no sistema como um todo. No caso do APA, a duplicação de diversos caminhos pode gerar conflitos, como visto na Seção 3.4, onde um caminho já duplicado não pode ser alvo da técnica novamente. Essa restrição limita a aplicação do APA que pode não apresentar boa cobertura em alguns circuitos onde muitos caminhos compartilham as mesmas portas lógicas.

A fim de quantificar essa aplicabilidade das técnicas, uma análise foi conduzida acerca do ganho que essas promovem em circuitos complexos. As principais medidas utilizadas para tal são:

- **Caminhos alvos:** Quantidade de caminhos lógicos selecionados para inserção das técnicas. Devido aos conflitos gerados por domínios duplicados dentro do circuito, nem todos os caminhos alvos são necessariamente beneficiados pelas técnicas, principalmente pelo APA;
- **Caminhos atingidos:** Número de caminhos críticos que obtiveram suas margens temporais efetivamente aumentadas pelo APA ou EST em, ao menos, 5%;
- **Cobertura:** Razão entre caminhos atingidos e alvos. Representa a capacidade das técnicas de abarcar o circuito em questão.

Esses valores foram estimados através da ferramenta de análise estática temporal (STA) do Cadence RTL Compiler, sendo uma aproximação dos valores que um circuito fabricado possuiria.

### 4.2.2 Resultados de Inserção

Os circuitos utilizados para medir a cobertura do APA são *benchmarks* empregados na competição do *International Symposium on Physical Design 2012 (ISPD12)*. Os circuitos são fornecidos já sintetizados em uma tecnologia específica para a competição e suas informações principais podem ser vistas na Tabela 8.

Tabela 8 – Circuitos utilizados para teste de cobertura com APA e EST

Nome	Entradas	Saídas	Total de Células
DMA	683	276	25301
pci_bridge32	160	201	33203
des_perf	234	140	111229
vga_lcd	85	99	164891
b19	22	25	219268
leon3mp	254	79	649191
netcard	1836	10	958780

Fonte: (OZDAL et al., 2012)

Cada circuito teve o número de caminhos críticos escolhido a partir do pior caso. Considerando que o pior caminho de cada circuito possui um atraso de propagação  $t_{pior}$ , são considerados críticos todos aqueles que possuem um atraso total  $t_{pd}$  tal que  $t_{pd} \geq t_{pior} * margem$  de forma similar ao Algoritmo 1 na Seção 3.5. As simulações foram conduzidas com valores de 75 %, 80 %, 85 % e 90 % de margem, com cada uma possuindo menos caminhos críticos alvos do que a anterior.

Além da margem de caminhos críticos, uma outra é estabelecida pelo Algoritmo 1 para a decisão de inserção do APA ou EST, chamada margem de inserção, ou simplesmente *mapa*. Os valores testados para essa margem foram de 70 %, 80 % e 90 %, com cada caso estabelecendo o limite mínimo entre atraso de propagação do pior caminho seguinte e o atraso do crítico que o antecede.

Para evitar os conflitos de domínios duplicados somente a duplicação parcial de caminhos foi empregada no APA. Essa foi configurada para realizar a duplicação das 10 primeiras portas lógicas dos caminhos que sucedem aos críticos. Conflitos de domínios ou caminhos com menos do que 10 portas, no entanto, podem reduzir esse número em casos específicos.

A Tabela 9 organiza os valores obtidos para uma margem de caminhos críticos (CC) de 75 % e *mapa* 70 %. A coluna CC informa a quantidade de caminhos críticos alvos de ambas as técnicas. Os números desses que se tornaram alvos do APA ou EST são mostrados nas colunas “Alvos APA” e “Alvos EST”, respectivamente. A quantidade de caminhos atingidos, isso é que obtiveram ao menos 5 % de ganho em margem temporal, para o APA e EST é mostrado nas colunas “Ating. APA” e “Ating. Est”, respectivamente.

A cobertura total das técnicas (CT) é a razão entre caminhos efetivamente atingidos por ambas e o número total de alvos CC. O custo de aplicação do APA pode ser estimado a partir das colunas de acréscimo de área e consumo. Essa estimativa é uma aproximação, já que a quantidade de *buffers* necessários para se gerar o empréstimo temporal e a janela de detecção do APA não está incluído no cálculo.

Tabela 9 – Resultados para inserção do APA e EST com 75 % de margem de CC e *mapa* 70 %

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	144	47	31	97	97	88,9%	0,3%	0,3%
des_perf	1460	915	34	545	312	23,7%	17,1%	16,8%
b19	717	161	19	556	465	67,5%	0,6%	0,6%
pci_bridge32	106	3	2	103	103	99,1%	0,1%	0,1%
netcard	1165	0	0	1165	1165	100,0%	0,0%	0,0%
leon3mp	8080	0	0	8080	8080	100,0%	0,0%	0,0%
DMA	87	2	1	85	85	98,9%	0,2%	0,2%

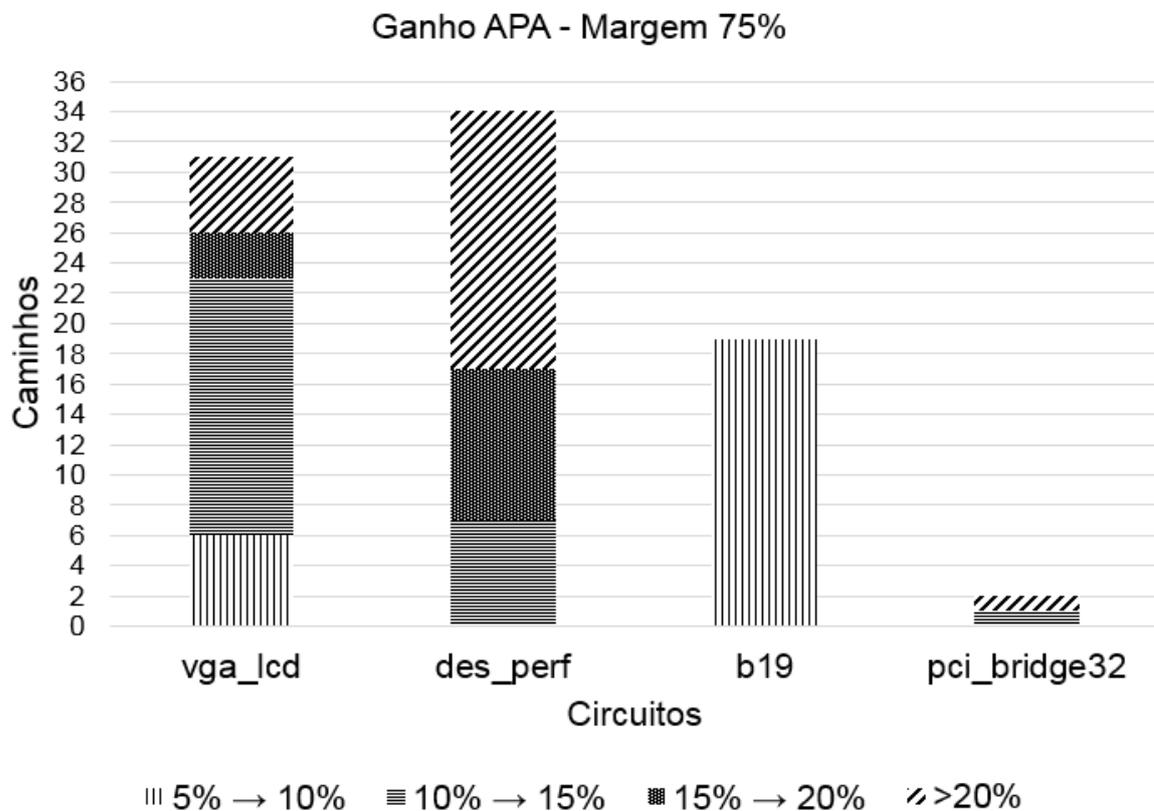
A forma como o fluxo foi implementado para os testes realiza a inserção do APA no maior número de caminhos alvos possível. O cálculo da janela de detecção e, conseqüentemente, do número de caminhos atingidos é realizado após a inserção foi concluída. Isso resulta no alto custo de área e consumo observado no “des\_perf” na Tabela 9, mesmo com poucos caminhos atingidos.

A Figura 4.15 mostra um gráfico de caminhos atingidos pelo APA para cada circuito de teste em que houve ao menos um. Os ganhos de margem temporal para cada caminho são representados pelos diferentes padrões no gráfico na forma de margens. O circuito “des\_perf”, por exemplo, possui um total de 34 caminhos atingidos pelo APA. Desses, 7 possuem ganho entre 10 % à 15 % (10 % → 15 %), 10 caminhos tiveram suas margens melhoradas entre 15 % à 20 % (15 % → 20 %) e 17 apresentaram mais de 20 % (> 20 %).

Devido a complexidade da técnica e do estágio inicial em que se encontra o fluxo de inserção, o APA apresentou poucos caminhos atingidos. No circuito “des\_perf”, por exemplo, de 915 alvos da técnica somente 34 foram efetivamente duplicados a ponto de gerar janelas de detecção com ganho acima de 5 %, resultando em uma cobertura de 3,7 %. Ao todo, cinco dos sete circuitos ISPD receberam o APA em alguma quantidade e, dentre eles, os que obtiveram melhor cobertura foram o “vga\_lcd” e “pci\_bridge32”, com 66,0 % e 66,7 % respectivamente. A cobertura do EST, por outro lado, é mais significativa, com circuitos alcançando 100 % de CT, como no caso do “leon3mp” onde todos os 8080 caminhos alvos foram atingidos.

No geral, as janelas de detecção criadas pelo APA fornecem um ganho de margem temporal menor do que 20 %, com muitos gerando janelas entre 10 % à 15 % para os

Figura 4.15 – Valores de ganho das janelas de detecção do APA com margem 75% de CC e *mapa* 70%



caminhos críticos que foram atingidos.

Os valores obtidos para margem de 90% podem ser observados na Tabela 10. A redução do número de caminhos críticos reduz também a quantidade dos alvos e atingidos pelas técnicas. Para o circuito com maior número de alvos, “des\_perf”, a quantidade de caminhos efetivamente atingidos pelo APA segue baixa, com somente 4,4% de cobertura específica da técnica e um valor CT de 57,2%.

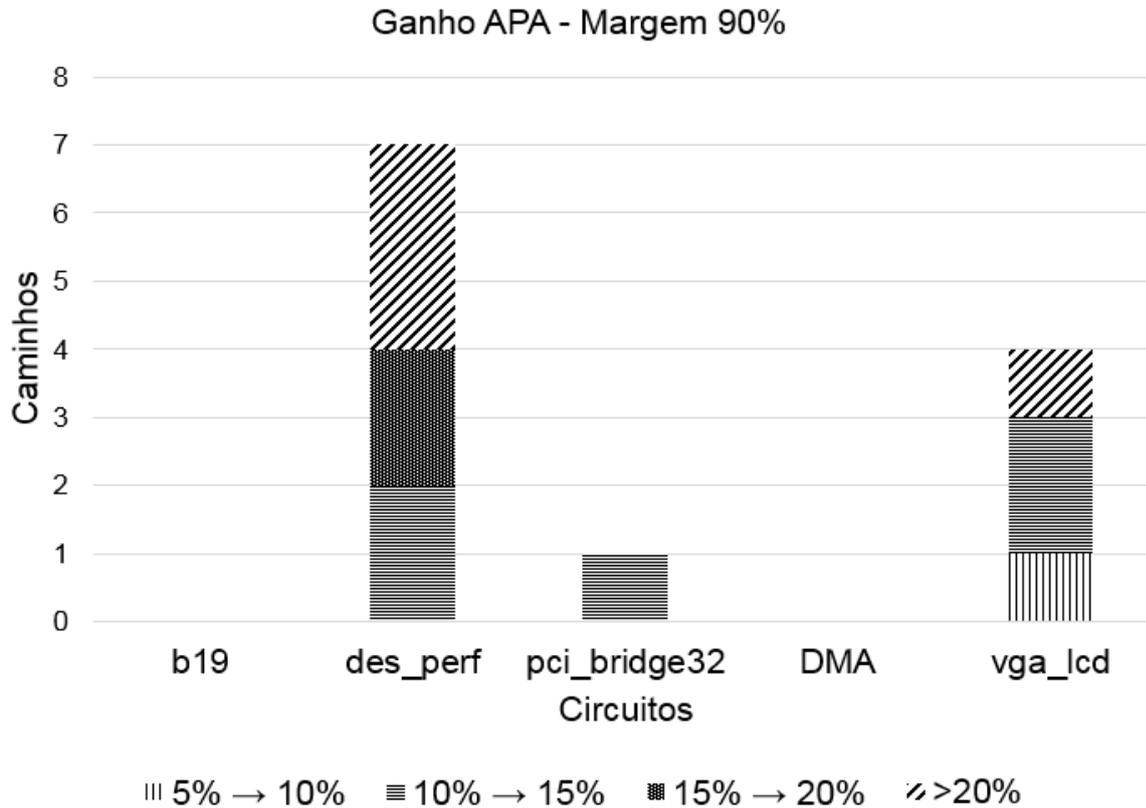
Tabela 10 – Resultados para inserção do APA e EST com 90% de margem e *mapa* 70%

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	21	4	4	17	17	100,0%	0,0%	0,0%
des_perf	402	158	7	244	223	57,2%	3,5%	3,4%
b19	146	14	0	132	132	90,4%	0,1%	0,0%
pci_bridge32	2	1	1	1	1	100,0%	0,1%	0,1%
netcard	261	0	0	261	261	100,0%	0,0%	0,0%
leon3mp	197	0	0	197	197	100,0%	0,0%	0,0%
DMA	10	0	0	10	10	100,0%	0,0%	0,0%

Os ganhos temporais do APA para a margem CC de 90% podem ser observados na

Figura 4.16. Como um todo, os ganhos obtidos são pequenos. Os circuitos “b19” e “DMA” não são atingidos pela técnica e somente 3 caminhos críticos tiveram janelas com ganho acima de 20 % no “des\_perf”.

Figura 4.16 – Valores de ganho das janelas de detecção do APA para 90 % de margem e *mapa* 70 %



O aumento do valor de *mapa* resulta na transferência de caminhos alvos do APA para o EST. Os resultados obtidos para um valor de margem 75 % de CC e *mapa* 90 % pode ser verificado na Tabela 11.

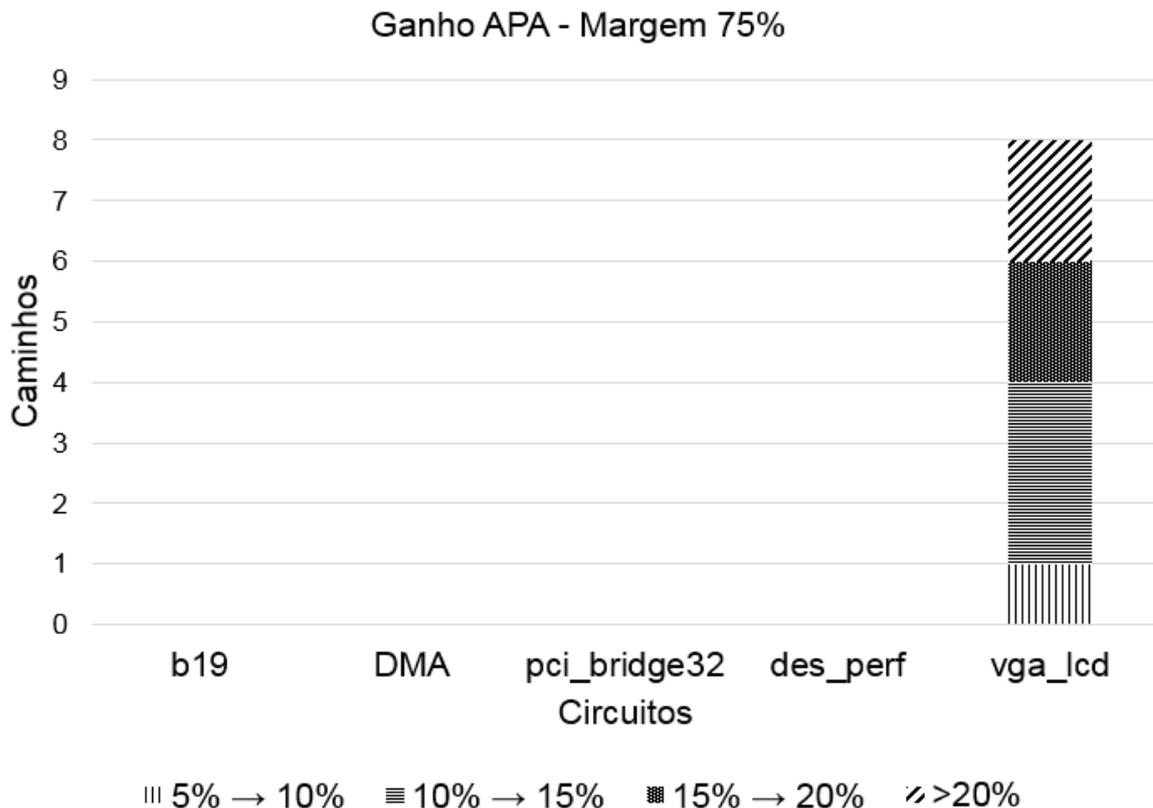
Tabela 11 – Resultados para inserção do APA e EST com 90 % de margem de CC e *mapa*

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	21	0	0	21	21	100,0%	0,0%	0,0%
des_perf	402	0	0	402	398	99,0%	0,0%	0,0%
b19	146	0	0	146	146	100,0%	0,0%	0,0%
pci_bridge32	2	0	0	2	2	100,0%	0,0%	0,0%
netcard	261	0	0	261	261	100,0%	0,0%	0,0%
leon3mp	197	0	0	197	197	100,0%	0,0%	0,0%
DMA	10	0	0	10	10	100,0%	0,0%	0,0%

O ganho obtido das janelas é mostrado na Figura 4.17. A redução do valor de *mapa* causou uma queda significativa no número de caminhos alvos do APA. Somente 8 caminhos

no circuito “vga\_lcd” foram selecionados para inserção do APA e, nesse caso, todos foram duplicados com sucesso. No entanto, somente dois caminhos no circuito apresentaram ganhos acima de 20% em margem temporal, em contraste com 136 caminhos atingidos pelo EST.

Figura 4.17 – Valores de ganho das janelas de detecção do APA para 90% de margem e *mapa*



As tabelas com valores para todas as margens simuladas podem ser consultadas no Apêndice A.

Os resultados obtidos das simulações de inserção apontam para uma cobertura significativa por parte do EST, com circuitos alcançando 100% quando o APA não era empregado juntamente à técnica, como no caso do circuito “leon3mp”. Essa técnica, apesar de já explorada na literatura de formas similares, apresenta a grande vantagem da simplicidade, focando somente na realocação temporal e não na detecção de erros.

Os valores obtidos do APA, no entanto, apontam para uma aplicação mais restrita. A técnica demonstrou ser demasiado complexa para a produção de ganhos consideráveis no estado atual de desenvolvimento. Em circuitos onde houve um número considerável de alvos, poucos caminhos foram efetivamente duplicados, resultando em baixa cobertura. A necessidade de se duplicar caminhos lógicos em circuitos complexos torna-a refém da complexidade e arranjo interno da lógica. Além disso, o estágio atual de desenvolvimento e avaliação da técnica pode não ser o suficiente para explorar todo o seu potencial.

As simulações realizadas utilizam aproximações para medir o ganho provido pelas técnicas. Por isso, faz-se necessária uma análise orientada à degradação e vida útil dos circuitos de teste para medir seu real benefício em termo de falhas ao longo da operação.

Uma análise de circuitos complexos utilizando frequência máxima peca ao simplificar os efeitos da degradação. A redução no período de *clock* força erros dinâmicos ao longo de todo o circuito de maneira uniforme em caminhos com temporizações próximas. A degradação, por outro lado, afeta diversas áreas de um chip com diferentes graus de intensidade, dependendo da temperatura, atividade e tensão de alimentação naquelas regiões.

Uma abordagem provável de render melhores resultados para o APA é unindo a sua inserção com uma análise de degradação do circuito alvo. Um novo fluxo que leve em conta informações não somente do atraso de caminhos críticos, mas também da probabilidade de falha e perfis de degradação das células pode direcionar a inserção para caminhos com maior chance de falha ao longo da vida útil do circuito.

A análise conduzida sobre o impacto das técnicas em área e consumo é limitada devido a natureza fictícia da biblioteca empregada. Os parâmetros utilizados por essas células podem não representar valores reais de células físicas e, por tanto, não são confiáveis para se investigar os custos das técnicas. Além disso, diferentes topologias de abordagens para produzir o sinal de *clock* atrasado podem ser empregadas, como a inserção de *buffers* em cada TBFF/APAFF contra a criação de uma árvore de distribuição central do sinal. Dependendo da topologia empregada, os impactos de área e consumo das técnicas variarão.

A forma como o fluxo foi implementado também dificulta uma verificação desses impactos. A inserção do APA é realizada no maior número de portas lógicas possíveis, respeitando o limite estabelecido pela duplicação parcial. Isso resulta em muitos caminhos alvos que tiveram a lógica seguinte duplicada mas que, devido a topologia do circuito, não produziram uma janela de detecção aceitável (nesse caso, acima de 5% de ganho), resultando em uma grande quantidade de portas duplicadas desnecessariamente. Esse comportamento acontece porque a estimação do tamanho das janelas só ocorre após as duplicações, caracterizando o fluxo atual como “cego” diante das alterações no instante em que elas estão sendo feitas.

Em um fluxo aplicado a um circuitor real, esse comportamento “cego” deve ser trocando por um algoritmo que monitore a criação das janelas de forma otimizada, realizando a duplicação somente quando essa gerar benefícios ao caminho crítico alvo.

## 4.3 Processador ARM

O desenvolvimento e simulação das técnicas rendeu um artigo científico publicado no trigésimo Simpósio de Projeto de Sistemas e Circuitos Integrados, SBCCI (*Symposium on Integrated Circuits and Systems Design*) realizado no *Chip on the Sands* em Fortaleza, Ceará. O artigo é intitulado *Mitigation of Aging Effects Through Selective Time-borrowing and Alternative Path Activation* e foi apresentado no mês de agosto de 2017.

Para verificar o funcionamento real das técnicas propostas, um processador com conjunto de instruções ARMv2, denominado AMBER, foi modificado com ambas. A Figura 4.18 mostra o *layout* do processador enviado para fabricação através do programa MOSIS. A tecnologia empregada é de 130nm e as bibliotecas de células foram fornecidas pelas companhias Global Foundries e ARM e o chip possui uma área total de 1,5x1,5mm<sup>2</sup>. Ao todo, o chip conta com 42 pinos para inserção e leitura de dados, alimentação e aterramento.

O processador pode ser encontrado nas versões Amber23 e Amber25 com *pipeline* de três e cinco estágios, respectivamente. A versão utilizada neste trabalho é o Amber23, e seu diagrama de *pipeline* é mostrado na Figura 4.19 composto por três estágios: *Fetch*, *Decode* e *Execute*. A memória *cache* de dados e instrução é unificada e do tipo associativa por conjunto, disponível nas configurações de 2, 3, 4 ou 8 vias de 4kB cada. A política de leitura e escrita de memória é do tipo “*write through*”. A comunicação externa do processador é realizada através de um barramento de 32 pinos com protocolo *wishbone* (PROJECT, 2013).

Ao todo, 67 caminhos críticos do AMBER foram modificados pelo APA e 74 pelo EST. Os acréscimos de área e potência foram estimados em cerca de 4,0% e 1,4%, respectivamente. O circuito do AMBER foi adaptado para conter uma entrada de sinal *clock* atrasado, além do original, que conecta todos os pinos de *clock* dos TBFF e APAFF. Com isso, espera-se observar o funcionamento do APA e EST com diferentes janelas de empréstimo temporal e de detecção de erros através da variação do atraso.

Outras alterações no circuito envolveram a serialização do barramento *wishbone*, para reduzir o número de pinos necessários para comunicação com o processador, e a redução do tamanho cache para uma totalmente associativa de 2kB, devido a área limitada do chip.

Figura 4.18 – Layout do microprocessador ARMv2-AMBER modificado pelo APA e EST

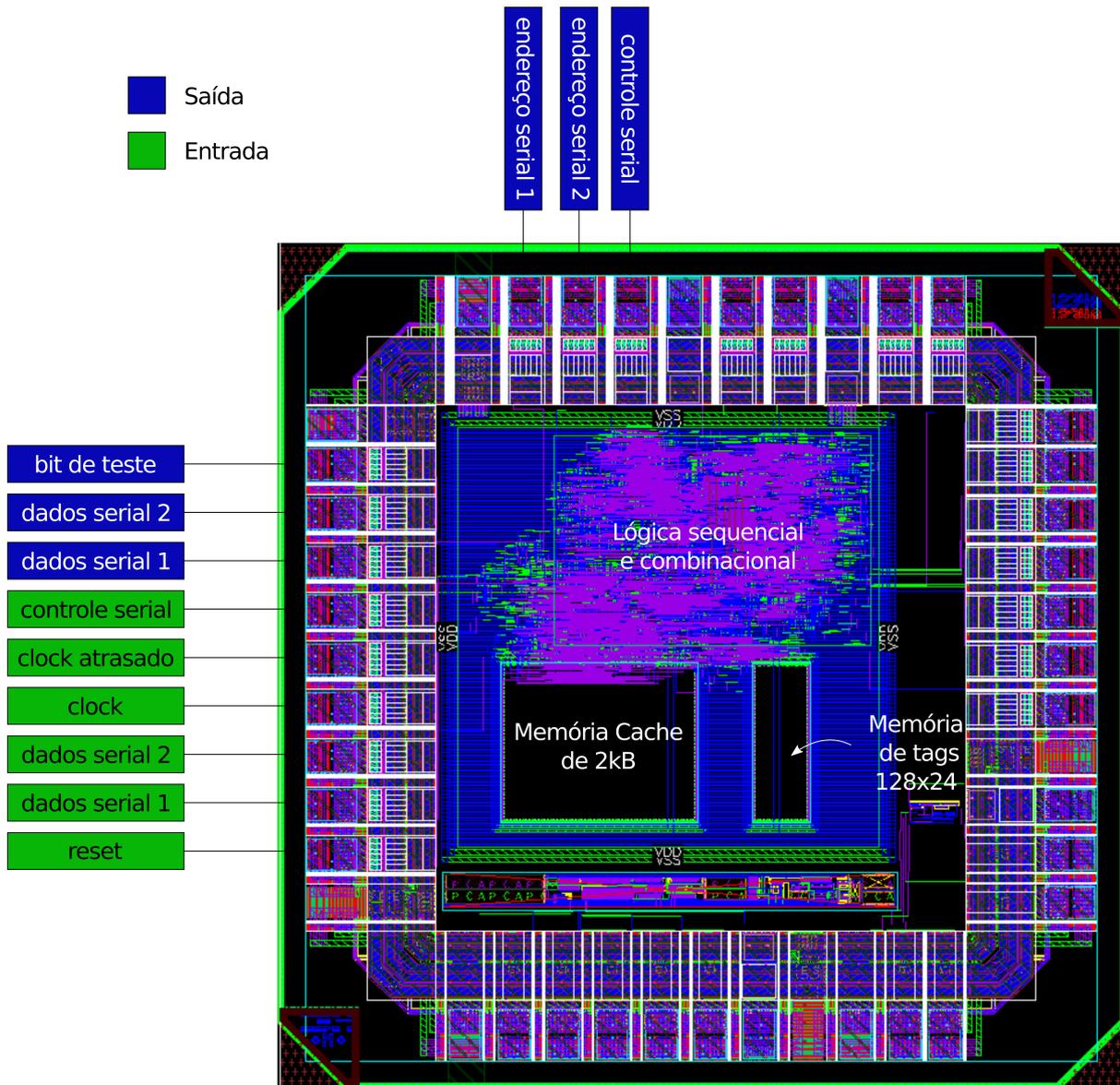
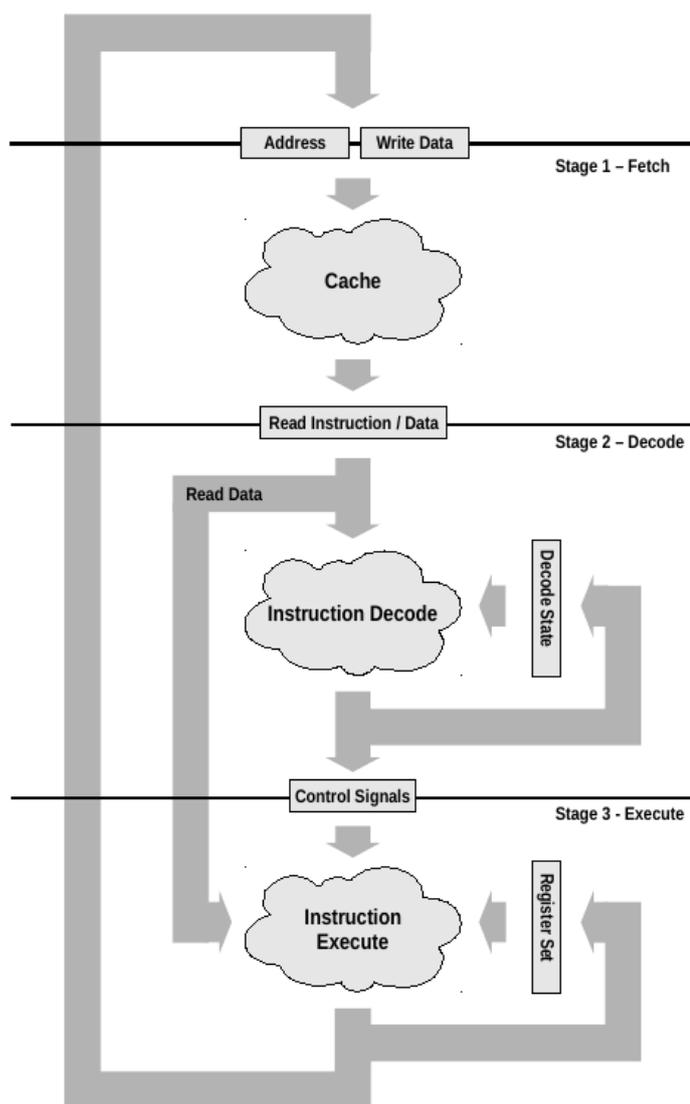


Figura 4.19 – Diagrama do processador AMBER



Fonte: Project (2013)



## 5 Conclusão

O rumo traçado pelo avanço da microeletrônica é marcado pelo aumento considerável de desempenho, funcionalidades e integração devido a intensa miniaturização dos transistores de efeito de campo FET. No entanto, essa miniaturização também trouxe problemas de robustez e confiabilidade para os circuitos integrados da atualidade. Esses problemas, acompanhados pela crescente ubiquidade de sistemas computacionais, culmina no aumento de demandas por circuitos e sistemas tolerantes a falhas.

Este trabalho faz uma revisão dos principais problemas que afligem sistemas computacionais digitais modernos e das técnicas de confiabilidade usualmente empregados na indústria para mitigar seus efeitos. O termo tolerância a falhas é empregado para indicar a capacidade de sistemas em continuar trabalhando mesmo na presença de defeitos físicos. Esses podem advir de variações no processo de fabricação ou na degradação devido a vários fenômenos físicos influenciados por condições ambientais.

Um foco principal foi dado na manifestação da degradação física na forma de aumento no atraso de células lógicas em circuitos digitais. Essas variações de atraso podem levar sistemas temporizados, como microprocessadores, a falhar por violação nas restrições temporais impostas durante o projeto do circuito. Para lidar com essas violações, denominadas erros temporais, técnicas de detecção e correção de erros *in situ* são encontradas na literatura, definindo um estado da arte extenso.

De forma a complementar as técnicas voltadas ao aumento de confiabilidade e tolerância a falhas, duas foram propostas pelo presente trabalho intituladas Empréstimo Seletivo Temporal (EST) e Ativação de Caminho Alternativo (APA). A primeira opera através da realocação de tempo entre caminhos lógicos, utilizando folgas temporais para aumentar a margem de processamento em caminhos críticos do circuito. A segunda atua quando essas folgas temporais não existem, ou seja, em caminhos críticos sucedidos por outros igualmente congestionados. Nesses casos o APA utiliza da duplicação dos caminhos lógicos seguintes ao crítico para criar uma janela de detecção e correção de falhas temporais.

As técnicas foram propostas tendo-se em mente o aumento da resiliência de circuitos integrados contra degradação. Efeitos transientes como interferência eletromagnéticas ou por radiação cósmica não foram abarcados e exigem outros tipos de soluções.

Ambas as técnicas foram elaboradas e simuladas buscando-se resultados que indicassem seu funcionamento e aplicabilidade em circuitos integrados digitais complexos.

Simulações de validação foram realizadas em nível elétrico, comprovando o funcionamento das técnicas em circuitos de teste simples. A métrica utilizada para essa avaliação

foi a frequência máxima em que o circuito consegue operar sem emitir resultados errôneos. Três circuitos de teste foram utilizados nessas simulações. Dois que utilizam cadeias de inversores para simular caminhos lógicos, intitulados INVCHAIN1 e INVCHAIN2. O terceiro é constituído por somadores do tipo *ripple-carry* em um arranjo de compressor 4:2, denominado COMPRESSOR. Ambos circuitos foram divididos em dois estágios sequenciais para possibilitar a aplicação das técnicas.

Resultados obtidos nessas simulações apontaram para ganhos em torno de 27% na frequência máxima dos INVCHAINS e em torno de 5,3% com o COMPRESSOR quando as técnicas foram aplicadas. O ganho obtido no COMPRESSOR foi limitado pelo segundo caminho crítico do circuito que, após a inserção das técnicas e aumento da frequência crítica, passou a ser o caminho mais congestionado do circuito. Essa simulação também serviu para exemplificar a operação de ambas as técnicas em sistemas com vários caminhos críticos, onde um pode assumir o papel de fonte de erros temporais quando outro é abarcado pelo EST ou APA.

Para testar a aplicação das técnicas em sistemas complexos, um fluxo automático de inserção foi proposto e testado em *benchmarks* do *International Symposium on Physical Design 2012*, ou ISPD12. Esses circuitos tiveram seus caminhos críticos modificados pelas técnicas com o objetivo de se analisar a aplicabilidade e benefícios destas. Os resultados obtidos apontam para ganhos significativos com o EST, incluindo boa cobertura dos caminhos alvos selecionados. O APA, por outro lado, apresentou resultados inferiores, tendo valores de cobertura muito baixas para ser de algum impacto positivo no circuito.

A metaestabilidade foi detectada como um problema para as técnicas quando essa se propagava do latch mestre para o escravo no flip-flop principal do APA. Para essa situação ocorrer, no entanto, o sinal deveria transicionar em uma janela de tempo com 20 a 30 ps de duração com a tecnologia utilizada. Nenhum circuito específico para tratamento de metaestabilidade foi testado devido a margem pequena de ocorrência desse fenômeno nas simulações conduzidas.

Circuitos integrados utilizados em aplicações críticas exigem que várias técnicas aumentem sua robustez para garantir a segurança das vidas e patrimônios envolvidos. As técnicas propostas mostraram, ao longo das simulações executadas, que são capazes de aumentar essa robustez através da expansão do tempo de processamento em caminhos críticos do circuito. O EST, principalmente, obteve aumentos significativos das margens temporais dos caminhos críticos. O APA demonstrou ser uma técnica de aplicabilidade limitada em circuitos complexos, apesar dos bons resultados obtidos nos testes com circuitos mais simples.

Apesar dos resultados negativos do APA, a técnica possui margem para otimização. Os conflitos de duplicação podem ser resolvidos utilizando algoritmos mais inteligentes e orientados à degradação das células e com informações de layout. Nesse caso, o APA

poderia ser aplicado de maneira orientada à posição dos caminhos críticos e não somente ao atraso.

## 5.1 Objetivos Alcançados

O presente trabalho conduziu uma pesquisa extensa na área de confiabilidade de circuitos integrados digitais e reuniu os principais conceitos utilizados na indústria e no estado da arte.

Duas técnicas foram propostas direcionadas para degradação de circuitos microeletrônicos. Essas foram testadas em diferentes ambientes de simulação e seu funcionamento foi comprovado nos circuitos utilizados. Através dessas simulações os ganhos das técnicas, medidos em aumento de margem temporal para caminhos críticos, foram estimados.

Um fluxo de inserção automático foi proposto e implementado para testes, culminando em um *software* escrito em linguagem de programação Tcl que realiza essa função. Apesar de seu desempenho requerer otimizações no que diz respeito ao APA, a inserção de ambas as técnicas já acontece de forma automática.

## 5.2 Trabalhos Futuros

As técnicas propostas foram desenvolvidas e exploradas em termo de viabilidade e aplicabilidade, mas muita margem existe para que trabalhos futuros possam expandir os conceitos elaborados. Um possível tópico de contribuição é a otimização das técnicas junto a elaboração de um programa prático para sua aplicação em circuitos.

Os fluxos empregados na inserção não realiza otimizações sobre as técnicas. As janelas de detecção dos APAFF, por exemplo, são definidas após a alteração do circuito, o que caracteriza o estágio atual do APA como uma técnica “cega”, isso é, o benefício só é quantificado após a alteração do circuito. Novas rotinas podem ser implementadas para levar em conta as janelas durante a duplicação dos caminhos lógicos, computado quais duplicações devem ser feitas prioritariamente para gerar janelas maiores.

O EST, apesar de não apresentar uma abordagem necessariamente inovadora, possui a vantagem de não produzir sinal de erro e por isso espera-se um impacto menor de área e consumo no circuito do que as técnicas pesquisadas. Essa implementação de empréstimo temporal é única até onde foi pesquisado na literatura. As cadeias de inversores utilizadas nele e no APA, no entanto, não são a melhor forma de se produzir um sinal de *clock* atrasado. Meios alternativos podem ser pesquisados, incluindo circuitos analógicos específicos para produção de atraso. Um exemplo de circuito assim é cadeias de inversores do tipo *current-starved* (DALLY; POULTON, 2008).

O processador modificado com as técnicas foi enviado para fabricação mas, até a data de escrita deste trabalho, não foi recebido para testes. Uma possível continuidade deste trabalho é a caracterização do processador AMBER em busca de resultados reais de um circuito com APA e EST fabricados. Programas específicos e *benchmarks* podem ser empregados para este fim, verificando se existe de fato aumento na robustez quando as técnicas estão ativadas.

Efeitos da metaestabilidade não foram observados durante as simulações dos circuitos de teste, tornando difícil avaliar o desempenho do APA em cenários onde esse fenômeno ocorre. Trabalhos futuros, utilizando modelos capazes de simular a metaestabilidade, podem focar neste aspecto e testar o APAFF e TBFF sob novas condições de operação para melhor antever seu comportamento.

## Referências

BECHADE, R. A. *Fast 4-2 carry save adder using multiplexer logic*. [S.l.]: Google Patents, 1998. US Patent 5,805,491. Citado na página 77.

BENSOUSSAN, A. Microelectronic reliability models for more than moore nanotechnology products. *Facta Universitatis, Series: Electronics and Energetics*, v. 30, n. 1, p. P–1, 2017. Citado na página 22.

BERNSTEIN, K. et al. *High speed CMOS design styles*. [S.l.]: Springer Science & Business Media, 1998. Citado 4 vezes nas páginas 23, 42, 44 e 61.

BLACK, J. R. Mass transport of aluminum by momentum exchange with conducting electrons. In: IEEE. *Reliability Physics Symposium, 1967. Sixth Annual*. [S.l.], 1967. p. 148–159. Citado na página 30.

BLUM, M.; WASSERMAN, H. Reflections on the pentium division bug. *IEEE Transactions on Computers*, IEEE, v. 45, n. 4, p. 385–393, 1996. Citado na página 33.

BORKAR, S. Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *Ieee Micro*, IEEE, v. 25, n. 6, p. 10–16, 2005. Citado na página 21.

BRUNVAND, E. *Digital VLSI chip design with Cadence and Synopsys CAD tools*. [S.l.]: Addison-Wesley, 2010. Citado 2 vezes nas páginas 24 e 80.

BURD, T. D. et al. A dynamic voltage scaled microprocessor system. *IEEE Journal of solid-state circuits*, IEEE, v. 35, n. 11, p. 1571–1580, 2000. Citado na página 49.

CANNIZZARO, M. et al. Saferazor: Metastability-robust adaptive clocking in resilient circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*, IEEE, v. 62, n. 9, p. 2238–2247, 2015. Citado 4 vezes nas páginas 54, 55, 58 e 87.

CHOUDHURY, M. et al. Timber: Time borrowing and error relaying for online timing error resilience. In: IEEE. *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010*. [S.l.], 2010. p. 1554–1559. Citado 3 vezes nas páginas 24, 54 e 58.

CONSTANTINESCU, C. Trends and challenges in vlsi circuit reliability. *IEEE micro*, IEEE, v. 23, n. 4, p. 14–19, 2003. Citado na página 33.

DALLY, W. J.; POULTON, J. W. *Digital systems engineering*. [S.l.]: Cambridge university press, 2008. Citado 2 vezes nas páginas 63 e 103.

DAS, S. et al. Razorii: In situ error detection and correction for pvt and ser tolerance. *IEEE Journal of Solid-State Circuits*, IEEE, v. 44, n. 1, p. 32–48, 2009. Citado 4 vezes nas páginas 24, 52, 53 e 57.

DRAKE, A. et al. A distributed critical-path timing monitor for a 65nm high-performance microprocessor. In: IEEE. *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*. [S.l.], 2007. p. 398–399. Citado na página 49.

- ERNST, D. et al. Razor: A low-power pipeline based on circuit-level timing speculation. In: IEEE. *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*. [S.l.], 2003. p. 7–18. Citado 7 vezes nas páginas 24, 50, 51, 52, 57, 71 e 72.
- FAIRES, J. D.; BURDEN, R. Numerical methods, brooks. *Cole, second ed., Pacific Grove, CA, USA*, 1998. Citado na página 80.
- FIROUZI, F. et al. Aging-and variation-aware delay monitoring using representative critical path selection. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, ACM, v. 20, n. 3, p. 39, 2015. Citado 3 vezes nas páginas 28, 31 e 49.
- FOJTIK, M. et al. Bubble razor: Eliminating timing margins in an arm cortex-m3 processor in 45 nm cmos using architecturally independent error detection and correction. *IEEE Journal of Solid-State Circuits*, IEEE, v. 48, n. 1, p. 66–81, 2013. Citado 2 vezes nas páginas 53 e 58.
- HENNESSY, J. L.; PATTERSON, D. A. *Computer architecture: a quantitative approach*. [S.l.]: Elsevier, 2011. Citado na página 21.
- HUARD, V.; DENAIS, M.; PARTHASARATHY, C. Nbti degradation: From physical mechanisms to modelling. *Microelectronics Reliability*, Elsevier, v. 46, n. 1, p. 1–23, 2006. Citado na página 24.
- INSTRUMENTS, N. *Redundant System Basic Concepts*. 2008. National Instruments Web Site. Disponível em: <<http://www.ni.com/white-paper/6874/en/>>. Acesso em: 21 out 2017. Citado na página 34.
- JING, J.; LIANG, L.; MENG, G. Electromigration simulation for metal lines. *Journal of Electronic Packaging*, American Society of Mechanical Engineers, v. 132, n. 1, p. 011002, 2010. Citado na página 30.
- JOHNSON, B. W. Fault-tolerant microprocessor-based systems. *IEEE Micro*, IEEE, v. 4, n. 6, p. 6–21, 1984. Citado na página 33.
- JOHNSON, B. W. *Design & analysis of fault tolerant digital systems*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1988. Citado 4 vezes nas páginas 34, 36, 37 e 38.
- JOHNSON, B. W. An introduction to the design and analysis of fault-tolerant systems. *Fault-tolerant computer system design*, ch, v. 1, p. 1–84, 1996. Citado 3 vezes nas páginas 27, 31 e 36.
- KURD, N. A. et al. A multigigahertz clocking scheme for the pentium (r) 4 microprocessor. *IEEE Journal of Solid-State Circuits*, IEEE, v. 36, n. 11, p. 1647–1653, 2001. Citado na página 61.
- LAVAGNO, L.; SCHEFFER, L.; MARTIN, G. *EDA for IC implementation, circuit design, and process technology*. [S.l.]: CRC press, 2006. Citado na página 46.
- MARICAU, E.; GIELEN, G. *Analog IC reliability in nanometer CMOS*. [S.l.]: Springer Science & Business Media, 2013. Citado 2 vezes nas páginas 28 e 29.

NAKAI, M. et al. Dynamic voltage and frequency management for a low-power embedded microprocessor. *IEEE journal of solid-state Circuits*, IEEE, v. 40, n. 1, p. 28–35, 2005. Citado na página 49.

OVERCLOCKING.GUIDE. *The risks of overclocking*. 2014. Overclocking.Guide Web Site. Disponível em: <<http://overclocking.guide/the-risks-of-overclocking/>>. Acesso em: 30 out 2017. Citado na página 30.

OZDAL, M. M. et al. The ispd-2012 discrete cell sizing contest and benchmark suite. In: ACM. *Proceedings of the 2012 ACM International Symposium on Physical Design*. [S.l.], 2012. p. 161–164. Citado na página 91.

PROJECT, A. O. S. *Amber 2 Core Specification*. 2013. OpenCores Site. Disponível em: <<http://opencores.org>>. Acesso em: 20 dez 2017. Citado 2 vezes nas páginas 97 e 99.

SCHRODER, D. K.; BABCOCK, J. A. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing. *Journal of applied Physics*, AIP, v. 94, n. 1, p. 1–18, 2003. Citado 2 vezes nas páginas 30 e 31.

SERLIN, O. Fault-tolerant systems in commercial applications. *Computer*, IEEE Computer Society Press, v. 17, n. 8, p. 19–30, 1984. Citado na página 22.

SERVICES, T. M. 2017. MOSIS Web Site. Disponível em: <<https://www.mosis.com/>>. Acesso em: 14 dez 2017. Citado na página 75.

SHOUMAN, M. L. Reliability of computer systems and networks: Fault tolerance, analysis, and design,. *Wiley-Interscience*, 2001. Citado na página 22.

SIEWIOREK, D.; SWARZ, R. *Reliable Computer Systems: Design and Evaluatuion*. [S.l.]: Digital Press, 1992. Citado 2 vezes nas páginas 22 e 23.

SORIN, D. J. Fault tolerant computer architecture. *Synthesis Lectures on Computer Architecture*, Morgan & Claypool Publishers, v. 4, n. 1, p. 1–104, 2009. Citado 7 vezes nas páginas 24, 27, 32, 33, 34, 35 e 37.

STATHIS, J. H. Physical and predictive models of ultrathin oxide reliability in cmos devices and circuits. *IEEE Transactions on Device and Materials Reliability*, IEEE, v. 1, n. 1, p. 43–59, 2001. Citado 2 vezes nas páginas 24 e 29.

TADROS, R. N. et al. A low-power low-area error-detecting latch for resilient architectures in 28-nm fdsoi. *IEEE Transactions on Circuits and Systems II: Express Briefs*, IEEE, v. 63, n. 9, p. 858–862, 2016. Citado na página 87.

TAKEDA, E.; SUZUKI, N.; HAGIWARA, T. Device performance degradation to hot-carrier injection at energies below the si-sio 2 energy barrier. In: IEEE. *Electron Devices Meeting, 1983 International*. [S.l.], 1983. p. 396–399. Citado 2 vezes nas páginas 24 e 28.

TAN, C.-J. et al. Clocking schemes for high-speed digital systems. *IEEE transactions on computers*, IEEE, v. 100, n. 10, p. 880–895, 1986. Citado 2 vezes nas páginas 39 e 42.

TANENBAUM, A. S. *Redes de Computadores*. [S.l.]: Pearson, 2011. Citado na página 38.

- TSCHANZ, J. et al. Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging. In: IEEE. *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*. [S.l.], 2007. p. 292–604. Citado na página 49.
- TU, K. Recent advances on electromigration in very-large-scale-integration of interconnects. *Journal of Applied physics*, AIP, v. 94, n. 9, p. 5451–5473, 2003. Citado na página 30.
- WANG, S.; CHEN, J.; TEHRANIPOOR, M. Representative critical reliability paths for low-cost and accurate on-chip aging evaluation. In: ACM. *Proceedings of the International Conference on Computer-Aided Design*. [S.l.], 2012. p. 736–741. Citado na página 61.
- WANG, W. et al. The impact of nbtj on the performance of combinational and sequential circuits. In: ACM. *Proceedings of the 44th annual Design Automation Conference*. [S.l.], 2007. p. 364–369. Citado na página 49.
- WESTE, N.; HARRIS, D.; BANERJEE, A. Cmos vlsi design. *A circuits and systems perspective*, v. 11, p. 739, 2005. Citado 6 vezes nas páginas 40, 41, 42, 45, 46 e 66.
- WOLF, W. *Modern VLSI design: IP-based design*. [S.l.]: Pearson Education, 2008. Citado na página 48.
- YE, Y. et al. Statistical modeling and simulation of threshold variation under random dopant fluctuations and line-edge roughness. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, IEEE, v. 19, n. 6, p. 987–996, 2011. Citado na página 23.

## APÊNDICE A – Tabelas de Medições

Este apêndice contém as tabelas com valores obtidos das simulações de inserção do APA nos circuitos do ISPD12 a título de consulta e verificação. Os gráficos foram apresentados e discutidos na Seção 4.

As Tabelas 12, 13, 14 e 15 mostram os valores obtidos da inserção do APA e EST em circuitos com margem de caminhos críticos 75 %, 80 %, 85 % e 90 %, respectivamente e *mapa* de 70 %.

Tabela 12 – Resultados para inserção do APA e EST com 75 %

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	144	47	31	97	97	88,9%	0,3%	0,3%
des_perf	1460	915	34	545	312	23,7%	17,1%	16,8%
b19	717	161	19	556	465	67,5%	0,6%	0,6%
pci_bridge32	106	3	2	103	103	99,1%	0,1%	0,1%
netcard	1165	0	0	1165	1165	100,0%	0,0%	0,0%
leon3mp	8080	0	0	8080	8080	100,0%	0,0%	0,0%
DMA	87	2	1	85	85	98,9%	0,2%	0,2%

Tabela 13 – Resultado de inserção para margem de 80 %

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	97	31	19	66	66	87,6%	0,2%	0,2%
des_perf	1446	906	34	540	311	23,9%	17,0%	16,7%
b19	391	25	0	366	327	83,6%	0,2%	0,2%
pci_bridge32	73	3	2	70	70	98,6%	0,1%	0,1%
netcard	1165	0	0	1165	1165	100,0%	0,0%	0,0%
leon3mp	7281	0	0	7281	7281	100,0%	0,0%	0,0%
DMA	30	0	0	30	30	100,0%	0,0%	0,0%

Os valores obtidos utilizando *mapa* 80 % para as mesmas margens de caminhos críticos são mostrados nas Tabelas 16, 17, 18 e 19.

Os resultados obtidos para *mapa* 90 % são mostrados nas Tabelas 20, 21, 22 e 23.

Tabela 14 – Resultado de inserção para margem de 85 %

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	41	14	10	27	27	90,2%	0,1%	0,1%
des_perf	978	542	25	436	296	32,8%	10,7%	10,6%
b19	148	14	0	134	132	89,2%	0,1%	0,0%
pci_bridge32	40	2	2	38	38	100,0%	0,1%	0,1%
netcard	1041	0	0	1041	1041	100,0%	0,0%	0,0%
leon3mp	7169	0	0	7169	7169	100,0%	0,0%	0,0%
DMA	25	0	0	25	25	100,0%	0,0%	0,0%

Tabela 15 – Resultado de inserção para margem de 90 %

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	21	4	4	17	17	100,0%	0,0%	0,0%
des_perf	402	158	7	244	223	57,2%	3,5%	3,4%
b19	146	14	0	132	132	90,4%	0,1%	0,0%
pci_bridge32	2	1	1	1	1	100,0%	0,1%	0,1%
netcard	261	0	0	261	261	100,0%	0,0%	0,0%
leon3mp	197	0	0	197	197	100,0%	0,0%	0,0%
DMA	10	0	0	10	10	100,0%	0,0%	0,0%

Tabela 16 – Resultado de inserção para margem de 75 % e *mapa* 80 %

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	144	9	9	135	135	100,0%	0,1%	0,1%
des_perf	1460	0	0	1460	1386	94,9%	0,0%	0,0%
b19	717	0	0	717	681	95,0%	0,0%	0,0%
pci_bridge32	106	0	0	106	106	100,0%	0,0%	0,0%
netcard	1165	0	0	1165	1165	100,0%	0,0%	0,0%
leon3mp	8080	0	0	8080	8080	100,0%	0,0%	0,0%
DMA	87	0	0	87	87	100,0%	0,0%	0,0%

Tabela 17 – Resultado de inserção para margem de 80 %

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	97	6	6	91	91	100,0%	0,1%	0,0%
des_perf	1446	0	0	1446	1374	95,0%	0,0%	0,0%
b19	391	0	0	391	361	92,3%	0,0%	0,0%
pci_bridge32	73	0	0	73	73	100,0%	0,0%	0,0%
netcard	1165	0	0	1165	1165	100,0%	0,0%	0,0%
leon3mp	7281	0	0	7281	7281	100,0%	0,0%	0,0%
DMA	30	0	0	30	30	100,0%	0,0%	0,0%

Tabela 18 – Resultado de inserção para margem de 85 %

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	41	4	4	37	37	100,0%	0,0%	0,0%
des_perf	978	0	0	978	957	97,9%	0,0%	0,0%
b19	148	0	0	148	146	98,6%	0,0%	0,0%
pci_bridge32	40	0	0	40	40	100,0%	0,0%	0,0%
netcard	1041	0	0	1041	1041	100,0%	0,0%	0,0%
leon3mp	7169	0	0	7169	7169	100,0%	0,0%	0,0%
DMA	25	0	0	25	25	100,0%	0,0%	0,0%

Tabela 19 – Resultado de inserção para margem de 90 %

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	21	0	0	21	21	100,0%	0,0%	0,0%
des_perf	402	0	0	402	398	99,0%	0,0%	0,0%
b19	146	0	0	146	146	100,0%	0,0%	0,0%
pci_bridge32	2	0	0	2	2	100,0%	0,0%	0,0%
netcard	261	0	0	261	261	100,0%	0,0%	0,0%
leon3mp	197	0	0	197	197	100,0%	0,0%	0,0%
DMA	10	0	0	10	10	100,0%	0,0%	0,0%

Tabela 20 – Resultados para inserção do APA e EST com 75 % de margem de caminhos críticos e *mapa* 90 %

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	144	8	8	136	136	100,0%	0,1%	0,1%
des_perf	1460	0	0	1460	1386	94,9%	0,0%	0,0%
b19	717	0	0	717	681	95,0%	0,0%	0,0%
pci_bridge32	106	0	0	106	106	100,0%	0,0%	0,0%
netcard	1165	0	0	1165	1165	100,0%	0,0%	0,0%
leon3mp	8080	0	0	8080	8080	100,0%	0,0%	0,0%
DMA	87	0	0	87	87	100,0%	0,0%	0,0%

Tabela 21 – Resultados para inserção do APA e EST com 80% de margem CC e *mapa* 90%

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	97	6	6	91	91	100,0%	0,1%	0,0%
des_perf	1446	0	0	1446	1374	95,0%	0,0%	0,0%
b19	391	0	0	391	361	92,3%	0,0%	0,0%
pci_bridge32	73	0	0	73	73	100,0%	0,0%	0,0%
netcard	1165	0	0	1165	1165	100,0%	0,0%	0,0%
leon3mp	7281	0	0	7281	7281	100,0%	0,0%	0,0%
DMA	30	0	0	30	30	100,0%	0,0%	0,0%

Tabela 22 – Resultados para inserção do APA e EST com 85% de margem CC e *mapa* 90%

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	41	4	4	37	37	100,0%	0,0%	0,0%
des_perf	978	0	0	978	957	97,9%	0,0%	0,0%
b19	148	0	0	148	146	98,6%	0,0%	0,0%
pci_bridge32	40	0	0	40	40	100,0%	0,0%	0,0%
netcard	1041	0	0	1041	1041	100,0%	0,0%	0,0%
leon3mp	7169	0	0	7169	7169	100,0%	0,0%	0,0%
DMA	25	0	0	25	25	100,0%	0,0%	0,0%

Tabela 23 – Resultados para inserção do APA e EST com 90% de margem de caminhos críticos e *mapa*

Nome	CC	Alvos APA	Ating. APA	Alvos EST	Ating. EST	CT	Área	Consumo
vga_lcd	21	0	0	21	21	100,0%	0,0%	0,0%
des_perf	402	0	0	402	398	99,0%	0,0%	0,0%
b19	146	0	0	146	146	100,0%	0,0%	0,0%
pci_bridge32	2	0	0	2	2	100,0%	0,0%	0,0%
netcard	261	0	0	261	261	100,0%	0,0%	0,0%
leon3mp	197	0	0	197	197	100,0%	0,0%	0,0%
DMA	10	0	0	10	10	100,0%	0,0%	0,0%