CENTRO UNIVERSITÁRIO FEI

AUGUSTO ERNI KLAUS EINSFELDT

FALHAS POR EVENTOS ÚNICOS EM MÁQUINAS DE ESTADO FINITAS:
Análise e proposta de arquitetura tolerante

São Bernardo do Campo

2019

AUGUSTO ERNI KLAUS EINSFELDT

FALHAS POR EVENTOS ÚNICOS EM MÁQUINAS DE ESTADO FINITAS:

Análise e proposta de arquitetura tolerante

Dissertação de Mestrado apresentada ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Mestre em Engenharia Elétrica, orientado pelo Prof. Dr. Renato Camargo Giacomini.

São Bernardo do Campo

2019

Einsfeldt, Augusto Erni Klaus.
FALHAS POR EVENTOS ÚNICOS EM MÁQUINAS DE ESTADO FINITAS: Análise e proposta de arquitetura tolerante / Augusto Erni Klaus Einsfeldt. São Bernardo do Campo, 2019.
101 p. : il.

Dissertação - Centro Universitário FEI.
Orientador: Prof. Dr. Renato Camargo Giacomini.

1. Radiação. 2. Confiabilidade. 3. Transiente. 4. Máquina de Estados. 5. Perturbação por Evento Único. I. Giacomini, Renato Camargo, orient. II. Título.

Elaborada pelo sistema de geração automática de ficha catalográfica da FEI com os dados fornecidos pelo(a) autor(a).

centro universitário FEI

APRESENTAÇÃO DE DISSERTAÇÃO
ATA DA BANCA EXAMINADORA

Programa de Pós-Graduação Stricto Sensu em Engenharia Elétrica

Mestrado

PGE-10

Aluno: Augusto Erni Klaus Einsfeldt **Matrícula:** 117301-2

Título do Trabalho: Falhas por eventos únicos em máquinas de estado finitas - análise e proposta de arquitetura tolerante.

Área de Concentração: Nanoeletrônica e Circuitos Integrados

Orientador: Prof. Dr. Renato Camargo Giacomini

Data da realização da defesa: 27/09/2019

ORIGINAL ASSINADA

Avaliação da Banca Examinadora:

_____

_____

_____

_____

_____

São Bernardo do Campo, / / .

MEMBROS DA BANCA EXAMINADORA	
Prof. Dr. Renato Camargo Giacomini	Ass.: _____
Prof. Dr. Rudolf Theoderich Buhler	Ass.: _____
Prof. Dr. Sandro Martini	Ass.: _____

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO ☒ REPROVADO ☐

VERSÃO FINAL DA DISSERTAÇÃO

APROVO A VERSÃO FINAL DA DISSERTAÇÃO EM QUE FORAM INCLUÍDAS AS RECOMENDAÇÕES DA BANCA EXAMINADORA

_____

Aprovação do Coordenador do Programa de Pós-graduação

_____

Prof. Dr. Carlos Eduardo Thomaz

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. Renato Camargo Giacomini e ao Prof. Dr. Rudolf Theoderich Bühler pelos ensinamentos e incansável apoio durante meus estudos.

Aos meus queridos filhos e filha cuja existência foi sempre motivação para meu crescimento pessoal e profissional.

Ao amigo Cláudio J. Richter pelas revisões e perguntas instigantes.

Agradeço a Profa. Luciene Rinaldi que me apresentou os professores de pós-graduação da FEI e incentivou para iniciar o curso de Mestrado na FEI.

Aos diretores da AS Avionics Services S.A. pelo incentivo e visão para a aplicação do conhecimento adquirido.

Ao Centro Universitário FEI nas figuras de seu corpo docente e administrativo que através do programa de bolsas e disponibilidade de infraestrutura educacional tornaram possível a realização deste curso.

Agradeço ainda a Cintia Fernandes, bibliotecária na FEI, que gentilmente revisou meu trabalho e deu importantes orientações.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

RESUMO

Este trabalho apresenta um estudo dos efeitos causados por partículas de alta energia, incidindo em dispositivos semicondutores, e as tecnologias aplicáveis para obter maior imunidade a isso nos circuitos eletrônicos. Motivado pela necessidade de projetar um circuito de controle para equipamento aeronáutico, além de considerar o modo de ação dos efeitos da radiação, foram desenvolvidas soluções de arquitetura digital para máquina de estado finitas programáveis. Ficou estabelecido que empregando técnicas de duplicação simples de registradores, verificação de integridade por paridade e distribuição dos sinais de *clock,* foi possível construir uma arquitetura para máquina de estados tolerante a perturbações por eventos únicos causadas por partículas de alta energia. A tecnologia resultante será empregada no desenvolvimento de dois equipamentos aeronáuticos. Sua versatilidade sugere que a continuação deste trabalho possa resultar no desenvolvimento de circuitos integrados próprios para uso em aplicações que exijam componentes de grande confiabilidade.

Palavras-chave: Radiação. Confiabilidade. Máquina de Estados. Eventos por Efeito Único. Transiente. Perturbação por Evento Único.

ABSTRACT

This work presents a study of the effects resulting from high energy particles striking semiconductor devices, along the applicable technologies, to obtain greater immunity in electronic circuits. Motivated by the design of a control circuit for an aeronautic equipment, along with the knowledge of the effects from radiation exposure, some solutions were developed for digital architectures applicable in programmable finite state machines. It was stablished that using techniques like register duplication, parity integrity verification and clock network distribution, it was possible to build an architecture for state machine, tolerant to single event effects derived from high energy particles. The resulting technology will be used in the design of two aeronautical equipment. Its versatility suggests that the continuation of this work may result in the development of integrated circuits designed for high reliability applications.

Keywords: Radiation. Reliability. State Machine. Single Event Effects. Transient. Single Event Upset.

LISTA DE ILUSTRAÇÕES

Figura 1 - Efeito da espessura da blindagem de Alumínio contra radiação 28
Figura 2 – Alteração de um bit de memória 30
Figura 3 - Cascata de partículas na atmosfera 31
Figura 4 - Reação nuclear no Boro-10 atingido por um Nêutron 32
Figura 5 - Interação de uma partícula de alta energia 34
Figura 6 - Corrente elétrica na coleta de cargas 34
Figura 7 - Esquema de um inversor 36
Figura 8 - Caminho da corrente elétrica induzida por uma partícula 37
Figura 9 - Descarga do capacitor de saída 38
Figura 10 - Célula de memória 38
Figura 11 - Leiaute de um transistor fechado 40
Figura 12 - Modelo ELT sob radiação 40
Figura 13 - Anel de proteção 41
Figura 14 - Esquema de uma memória DICE 43
Figura 15 - Circuito típico para um TMR 44
Figura 16 - Confiabilidade de circuitos NMR 46
Figura 17 - Circuito de um DTMR 47
Figura 18 - Máquina de Estados Finitos 48
Figura 19 - Máquina de Estados Programável 49
Figura 20 - Estrutura básica de um FPGA 52
Figura 21 – FCU, ACCU e a aeronave B-250 56
Figura 22 - Bancada de testes do FCU 57
Figura 23 - Arquitetura da PSM 60
Figura 24 - Barramento da memória ROM 62
Figura 25 - Barramento de endereço 63
Figura 26 – Representação da distribuição dos *clocks* defasados 65
Figura 27 – Uso de *clock enable* 67
Figura 28 - Barramentos de dados 68
Figura 29 - Esquema de recuperação de dados 71
Figura 30 - Esquema de um *flip-flop* da plataforma de teste 72

Figura 31 - PSM sem os circuitos de tolerância a falhas 73
Figura 32 - Compilador para a PSM 75
Figura 33 - Circuito para detecção e correção de falhas por SET ou SEU 81
Figura 34 - Diagrama de tempo 84
Figura 35 - Redundância de registrador com seletor único 86

LISTA DE TABELAS

Tabela 1 – Fluxo de Nêutrons em diversas localidades 25
Tabela 2 - Operações da ALU 68
Tabela 3 – Condições para seleção de endereço 69
Tabela 4 – Resultados dos testes em FPGA 79
Tabela 5 – Recursos ocupados em cada implementação 87

LISTA DE ABREVIATURAS E SIGLAS

ACC	*Accumulator* – Registrador acumulador de dados em uma CPU
ACCU	*Air-Conditioning Control Unit* – Unidade de controle do ar-condicionado
ALU	*Arithmetic-Logic Unit* – Unidade lógica e aritmética
ANAC	Agência Nacional de Aviação Civil
ARINC	*Aeronautical*, *Radio Incorporated* – Padrão de comunicação entre instrumentos empregado em sistemas aeronáuticos
BGA	*Ball Grid Array* – Encapsulamento empregando uma grade de esferas
BPSG	Vidro de Borofosfosilicato
CAST	*Certification Authorities Software Team* – Grupo de Software para autoridades certificadoras
CERN	*Conseil Européen pour la Recherche Nucléaire* - Organização Europeia para a Pesquisa Nuclear
CMOS	*Complementary Metal Oxide Semiconductor* – Semicondutor metal-óxido complementar
COTS	*Commercial Off-The-Shelf* - Item comercialmente disponível
CPU	*Central Processing Unit* – Unidade de processamento central
DDD	*Displacement Damage Dose* – Dose para dano por deslocamento
DTMR	*Delayed Tripple Modular Redundancy* – Redundância modular tripla com atrasos
EEPROM	*Electricaly-Erasable Programmable Read-Only Memory* – Memória ROM que pode ser apagada e reescrita eletricamente
EG	*Enclosed Gate* – Porta perifericamente fechada
ELT	*Enclosed Layout Transistor* - Transistor com desenho protegido
EMI	*Electromagnetic Interference* – Interferência eletromagnética
FAA	*Federal Aviation Administration* – Administração Federal da Aviação
FCU	*Flap Control Unit* – Unidade de controle de Flaps

FIT	*Failure In Time* – Falha ao longo do tempo
FLASH	Tecnologia de memória funcionalmente similar à EEPROM
FPGA	*Field Programmable Gate Array* - Matriz de portas digitais programável em campo
FSM	*Finite State Machine* – Máquina de estados finita
LATS	*Latin-America Test Symposium* – Simpósio sobre Testes realizado na América Latina
LEO	*Low Earth Orbit* – Órbita terrestre baixa
LET	*Linear Energy Transfer* - Transferência linear de energia
MBU	*Multiple Bit Upset* – Perturbação em múltiplos bits
MOSFET	*Metal-Oxide Semiconductor Field Effect Transistor* – Transistor de efeito de campo em tecnologia de semicondutor empregando Metal-Óxido
MTBF	*Mean Time Between Failures* – Tempo médio entre falhas
NIEL	*Non Ionizing Energy Loss* – Perda de energia não ionizante
NMOS	N-*channel* MOS – Transistor MOSFET de canal N
NMR	N-*Modular Redundancy* – Redundância N-Modular
OTP	*One-Time Programmable* – Memória que é fabricada apagada e que pode ser gravada uma única vez pelo usuário
PMOS	P-*channel* MOS – Transistor MOSFET de canal P
PNPN	Estrutura parasitária que ocorre em dispositivos semicondutores formando um arranjo equivalente a um Tiristor
PSM	*Programmable State Machine* - Máquina de estados programável
RAM	*Random Access Memory* – Memória de acesso aleatório
RC	Resistor Capacitor – Circuito de um resistor e um capacitor
RHBD	*Radiation Hardening By Design* - Desenho para reforço contra radiação
ROM	*Read-Only Memory* – Memória de apenas leitura
SEB	*Single Event Burnout* – Queima causada por evento único
SEE	*Single Event Effects* - Efeitos por Evento Único

SEFI	*Single Event Functional Interrupt* – Interrupção do funcionamento por causa de um evento único
SEGR	*Single Event Gate Rupture* – Ruptura da porta causada por um evento único
SEH	*Simple Electronic Hardware* – Circuito eletrônico simples
SEL	*Single Event Latchup* – Travamento da estrutura PNPN em decorrência de um evento único por radiação
SER	*Soft Error Rate* – Taxa de erros recuperáveis
SET	*Single Event Transient* - Transiente por Evento Único
SEU	*Single Event Upset* - Perturbação por Evento Único
STE	*Single Turbine Engine* – Avião com apenas um motor, tipo turbina
SRAM	*Static Random Access Memory* – Memória estática de acesso aleatório, permite escrita e leitura
TID	*Total Ionizing Dose* – Dose ionizante total
TMR	*Tripple Modular Redundancy* – Redundância modular tripla
TNID	*Total Non-Ionizing Dose* – Dose não ionizante total
TSC	*Totally Self-Checking* – Completamente auto verificável
ULA	*Ultra-Low Alpha* – Partícula Alpha de ultra-baixa energia

LISTA DE SÍMBOLOS

μ	Mobilidade [$cm^2/V.s$]
A	Número de Avogrado [$6{,}0221409 \times 10^{23}$]
E_{Field}	Campo elétrico [V]
I	Corrente elétrica [A]
I_{dreno}	Corrente de dreno [A]
L	Profundidade da coleta de cargas [µm]
LET	Energia linear transferida [$MeV \cdot cm^2/mg$]
LET_{th}	Limiar da energia linear transferida [$MeV \cdot cm^2/mg$]
P+	Região de um semicondutor fortemente dopada, tipo P
Q_{coll}	Carga coletada [fC]
Q_{crit}	Carga crítica [fC]
t_{th}	Tempo de comutação [s]
V_{th}	Tensão de limiar em um transistor MOS [V]
σ	*Cross-section* – Secção de choque [cm^2]
$\tau\alpha$	Tempo para coleta de cargas dependente da tecnologia [s]
$\tau\beta$	Tempo para ocorrer o funil ionizado [s]

SUMÁRIO

1 INTRODUÇÃO 18
2 CONCEITOS BÁSICOS 22
2.1 Efeitos por Evento Único (Single Event Effects – SEE) 22
2.2 Taxa de Erros 23
2.3 Efeitos da radiação ionizante em materiais semicondutores 25
2.4 Falhas permanentes 26
2.5 Fontes de radiação que causam falhas não permanentes 29
2.6 Transferência Linear de Energia 33
2.7 Efeitos da interação das partículas energéticas em transistores 33
2.8 Técnicas para Reduzir a Sensibilidade e Efeitos causados por SEE 39
2.9 RHBD por modificação do leiaute 39
2.9.1 ELT 39
2.9.2 Guard Ring 41
2.9.3 Ajuste dimensional 41
2.10 Arquiteturas Digitais 42
2.11 Redundância 43
2.12 Máquina de Estados Finita 47
2.13 Máquina de Estados Finita com maior confiabilidade 50
2.14 FPGA (Field Programmable Gate Array) 51
3 CRITÉRIOS PARA PROJETO DE EQUIPAMENTO AERONÁUTICO 54
4 DESENVOLVIMENTO DE SOLUÇÕES 59
4.1 Arquitetura PSM tolerante a falhas por SEU e SET 59
4.1.1 Caso especial: Inicialização 70
4.1.2 Recuperação de Dados no Acumulador 71
4.1.3 Verificação da Tolerância da Arquitetura Proposta 72

4.1.4 Compilador de linguagem 74
4.1.5 Método de teste 76
4.1.6 Resultados dos Testes 78
4.1.7 Considerações sobre Melhorias 79
4.2 Registrador Auto Recuperável Empregando Amplificador *Tri-State* 80
4.3 Registrador Auto Recuperável Empregando Seletor Único 85
4.4 Análise das arquiteturas de registradores auto recuperáveis 87
5 CONCLUSÃO 88
REFERÊNCIAS 90
APÊNDICE A – TRABALHOS PUBLICADOS 95

1 INTRODUÇÃO

O contínuo desenvolvimento tecnológico na fabricação de circuitos integrados tem proporcionado vantagens crescentes na redução de custos, no menor consumo de energia e na diminuição das dimensões físicas dos dispositivos. Notadamente, o maior avanço tecnológico tem sido nesta redução das dimensões dos dispositivos semicondutores que, por muitos anos, seguiu aproximadamente a conhecida Lei de Moore (Kumar, 2015). Nela afirma-se que o desenvolvimento continuado das tecnologias de semicondutores permitiria a duplicação da quantidade de transistores nos circuitos integrados a cada 2 anos aproximadamente. Nesse ritmo de integração de dispositivos, os transistores utilizados precisam ser projetados em tamanhos cada vez menores e a redução de tamanho os torna mais sensíveis a efeitos físicos que antes eram pouco frequentes.

Um destes efeitos físicos é a perturbação elétrica causada pela transferência de energia de uma partícula ionizante que atinja o dispositivo semicondutor. Esta perturbação faz parte dos efeitos conhecidos como Efeitos por Eventos Únicos (*Single Event Effects*, SEE). Dentre estes efeitos, a Perturbação por Evento Único (*Single Event Upset,* SEU) geralmente resulta numa falha registrada, porém temporária, também conhecida como *Soft Error*, que pode durar algum tempo até ser resolvida pelo próprio circuito eletrônico. Por exemplo, um evento causando uma mudança no conteúdo de uma memória que pode ser eventualmente reescrita e desse modo corrigida. Um outro tipo muito comum é chamado de Transiente por Evento Único (*Single Event Transient*, SET) que, como seu nome indica, causa uma interferência apenas transitória e que somente vai causar problemas se ocorrer em partes e momentos muito específicos em um circuito integrado.

Estas partículas ionizantes provêm de raios cósmicos e raios solares incidindo sobre o componente eletrônico, por produção secundária de partículas na atmosfera terrestre, ou ainda por fontes radioativas próximas (Wang e Agrawal, Vishwani D., 2008).

Circuitos eletrônicos em satélites e sondas espaciais, que estão constantemente expostos à radiação cósmica, eventualmente param de funcionar temporariamente e, em alguns casos, permanentemente. Isso é especialmente crítico em equipamentos aviônicos, cujo mal funcionamento poderia pôr vidas humanas em risco. Nas altitudes típicas de voo para aviões comerciais, acima de 10.000 metros, existem mais partículas energéticas porque a atmosfera é

mais rarefeita e não oferece um bom efeito de blindagem contra radiação (Youngquist *et al.*, 2014). Nesta altitude e acima dela também ocorre a geração secundária de partículas causada pelo impacto de Nêutrons e Prótons, de origem cósmica, em núcleos de átomos presentes na atmosfera.

A motivação para o estudo aqui apresentado aconteceu durante o desenvolvimento, pelo autor, de um equipamento aeronáutico destinado ao controle da posição de Flaps (superfícies de voo auxiliares) em um avião acrobático de alta performance. Como uma falha neste equipamento pode pôr em risco a integridade física dos tripulantes e pessoas em solo, as autoridades certificadoras exigem que o projeto eletrônico além de ser confiável, seja também seguro.

Apesar das diferenças de definição entre estes dois conceitos similares, onde um sistema confiável não implica que seja seguro e vice-versa (Mulazzani, 1985), para um Controlador de Flaps oferecer segurança é preciso que seja também confiável. Não basta o equipamento posicionar corretamente estas superfícies de voo (ser confiável). Elas também não podem ser movidas para uma posição diferente daquela esperada pelo comando do piloto (ser seguro). Para este avião especificamente, classificado como Classe 2, STE (*Single Turbine Engine*), de acordo com a AC-23.1309 da FAA (*Federal Aviation Administration*) (FAA, 2011), a probabilidade de falha para tal equipamento deve ser menor que uma em 10^5 horas de voo. Esta falha seria tanto por deixar de mover os Flaps quando solicitado, quanto por mover para uma posição diferente da esperada.

Nos circuitos eletrônicos do Controlador de Flaps isto significa que, além da funcionalidade permanecer sempre a mesma, independente da ocorrência de eventos únicos, os valores dos dados tratados têm que permanecer corretos. Por isso é importante que a mudança no valor de uma memória seja evitada, ou detectada e corrigida antes que se torne uma falha de segurança. O projeto, então, deve considerar a probabilidade de falhas de todos os componentes eletrônicos envolvidos, além da taxa de falhas causadas por SEU ou SET, para atender ao requisito de 10^5 horas de voo sem falhas.

Portanto, torna-se necessário desenvolver técnicas de projeto de circuitos que minimizem ou eliminem as falhas causadas por SEE. Estas técnicas podem envolver diversos aspectos como o desenho dos dispositivos em circuitos integrados de forma a mitigar os efeitos da ionização; emprego de escudos metálicos como blindagem cobrindo os circuitos;

redundância de circuitos; emprego de métodos de verificação e correção de erros; dentre outros (Trivedi e Mehta, 2016).

Em muitos sistemas críticos, emprega-se a técnica de redundância modular tripla (TMR), na qual os circuitos mais importantes são triplicados para garantir que a qualquer momento, a falha de um deles pode ser coberta pelos outros dois que apresentarão os mesmos valores de saída (Balasubramanian, Prasad e Mastorakis, 2016). Nessa técnica, um circuito de arbitragem determina a validade do valor apresentado pelos circuitos redundantes. Apesar do aumento de confiabilidade oferecido pela técnica TMR, esta não é totalmente imune a falhas e se o arbitrador não for também triplicado pode ser ainda mais susceptível a erros durante um evento único. Além disso, o uso de TMR implica em aumento na área de Silício em 200% ou mais (Balasubramanian, Prasad e Mastorakis, 2016) e consequentemente aumento da energia utilizada, limitando sua aplicação em alguns casos.

Uma outra solução, quando a funcionalidade do sistema não é crítica, é reinicializar o componente após uma falha ser detectada. Em um satélite, por exemplo, os sistemas de comunicação podem ser reinicializados após um erro ser detectado com o prejuízo apenas da interrupção momentânea dos serviços. Porém, o sistema de controle de atitude (orientação em relação a um referencial inercial), que mantém o satélite na orientação correta em relação à Terra, não seria recomendável apenas depender de reinicialização. Na eventualidade de um SEE, os circuitos deste sistema não devem poder disparar incorretamente os propulsores que controlam a atitude, a ponto de tirar o satélite de sua órbita ou deixa-lo inoperante.

Nesta perspectiva, este estudo vai analisar os efeitos de partículas de alta energia em dispositivos semicondutores e as consequências da ocorrência de SEU ou SET, quando estes dispositivos formam um circuito de memória, ao mudar o valor da informação armazenada. A seguir, são apresentados os efeitos em circuitos digitais mais complexos, como as máquinas de estado finitas e as técnicas de mitigação correntemente empregadas, sugerindo por fim uma arquitetura de máquina de estados finita que pode ser imune à SEU e SET. Será demonstrado que o emprego de técnicas de temporização em registradores, a duplicação de apenas alguns elementos da lógica e empregando verificação da validade dos dados armazenados, permitem obter tolerância através da recuperação do estado correto do sistema após a ocorrência de um evento único.

Simulações e testes práticos foram elaborados para comprovar esta tolerância e permitir determinar a confiabilidade do sistema. Como resultado, a tecnologia proposta e

desenvolvida ao longo deste estudo foi aplicada com sucesso em uma unidade de controle de Flaps e num controlador de ar-condicionado para aeronaves. Adicionalmente, esta tecnologia pode ser eventualmente utilizada também na elaboração de projetos de circuitos integrados digitais específicos para aplicações críticas, assim como no desenvolvimento de sistemas discretos empregando circuitos integrados convencionais.

2 CONCEITOS BÁSICOS

Este capítulo apresenta a fundamentação para o estudo dos efeitos da radiação ionizante sobre dispositivos semicondutores; a abrangência destes efeitos em circuitos de memória e nos circuitos digitais complexos, assim como as técnicas empregadas atualmente para mitigar estes efeitos.

2.1 Efeitos por Evento Único

Os efeitos por eventos únicos são chamados deste modo por serem causados por uma partícula que atinge um dispositivo semicondutor. Os efeitos são notáveis como na alteração de dados em memórias de computador, variação de parâmetros elétricos em circuitos analógicos ou mesmo destruição de componentes eletrônicos.

Os SEE são classificados em diversos tipos (LaBel, 1996) :

Single Event Upset (SEU) – Trata-se de uma mudança no estado de uma memória induzido por uma partícula energética. Considerado um erro recuperável (*soft error*) porque a reinicialização do sistema ou a reescrita do valor da memória permite o sistema voltar ao funcionamento normal.

Single Event Transient (SET) – Similar ao SEU porém ocorre como um transiente elétrico em um dispositivo sensível que pode causar mudanças no estado lógico em um ou mais circuitos. SETs em dispositivos lineares diferem significativamente de outros tipos de SEE como o SEU em uma memória (Adell *et al.*, 2000).

Single Event Latchup (SEL) – Uma condição que causa a perda funcional de um dispositivo devido à altas correntes resultantes. Um SEL ocorre pela ativação da estrutura parasitária PNPN (ocorre entre o substrato e elementos de transistores MOSFET, comum em estágios amplificadores de circuitos CMOS) e pode ou não danificar um componente permanentemente, mas exige no mínimo o desligamento da energia para permitir que a corrente cesse e o dispositivo volte ao estado normal.

Single Event Burnout (SEB) – Um efeito que destrói o componente, tipicamente ocorrendo em transistores de potência, também pela ativação da estrutura parasitária PNPN

localizada próximo do ponto de impacto. Em dispositivos operando em tensões altas ocorre um aumento súbito de corrente e temperatura por efeito Joule. Acontece principalmente em transistores MOSFET, mas já foi percebido em transistores bipolares. Transistores fabricados para funcionar abaixo de 100V são mais resistentes a este tipo de falha (Miller *et al.*, 2006). Manter a porta polarizada negativamente também ajuda a reduzir os riscos de SEB.

Single Event Gate Rupture (SEGR) – Neste caso a partícula energética consegue formar um caminho condutor no óxido de porta de um transistor MOSFET, causando redução na isolação da porta e eventualmente um curto-circuito entre a porta e o dreno ou fonte do transistor.

Multiple Bit Upset (MBU) – Um evento causado por uma única partícula, mas que causa a mudança de estado em múltiplos bits de uma memória ou circuito lógico.

Cada SEE tem uma característica exclusiva como polaridade, forma de onda, amplitude, duração, etc. Estas características dependem da localização do impacto das partículas, energia de partículas, tecnologia do dispositivo, tensão de alimentação do dispositivo e carga de saída (Wang e Agrawal, Vishwani D, 2008).

2.2 Taxa de Erros

A taxa de erros de um circuito causados por partículas energéticas (*Soft Error Rate*, SER) é medido empregando uma unidade que representa a taxa de falhas num intervalo de tempo (*Failure In Time*, FIT). A unidade FIT é utilizada para cálculo de confiabilidade de componentes e equipamentos, eventualmente resultando na medida de MTBF (*Mean Time Between Failures*). O MTBF equivale a 1/FIT e é utilizado para calcular quantas peças sobressalentes podem ser necessárias para se ter à disposição e assim garantir que um sistema funcione durante um certo tempo. A medida MTBF é fundamental no projeto e comercialização de equipamentos que precisem estar funcionando sempre que necessários.

Um FIT corresponde a uma falha em 10^9 horas. Empregando a quantidade de partículas que atingem uma determinada área do material (denominado de fluência) e conhecendo a quantidade de erros que resultam desta incidência é possível calcular a confiabilidade de um circuito em relação à ocorrência de SEE.

Considerando que, com uma fluência total N=partículas/cm^2, um certo número de erros *n* ocorrem em um determinado dispositivo, obtemos a secção de choque (*cross section*): $\sigma = n/N$, que representa a área em que deve ocorrer uma falha para este dispositivo.

Agora, considerando o fluxo de partículas existente em um determinado ambiente, dado por $\Phi = n/cm^2/h$, obtemos a equação (1). Ela representa a quantidade de falhas esperada para um dispositivo, quando exposto a um certo fluxo de partículas, durante um determinado tempo.

$$FIT = n/N * \Phi * 10^9 \tag{1}$$

Como um exemplo de cálculo, uma memória de 1 Mb (10^6 bits), com secção de choque σ de $5x10^{-14}$ cm^2 por bit, submetida a um fluxo de 5 Nêutrons/cm^2/h (fluxo teórico de Nêutrons na cidade de Nova Iorque ao nível do mar (Ziegler, 1996)) , resulta numa taxa de erros de 250 FIT, equivalente a uma falha em 1 bit a cada 4 milhões de horas de uso (Gaillard, 2011). A energia destas partículas não entra no cálculo porque considera-se que o *cross section* foi calculado empregando as mesmas partículas existentes no ambiente em que o circuito é depois exposto.

A mesma memória posicionada a uma altitude de 10.000 metros sobre a cidade de São Paulo estaria exposta a um fluxo teórico de 475 Nêutrons/cm^2/h (valor obtido em http://www.seutest.com/cgi-bin/FluxCalculator.cgi em Junho/2019) e resultaria numa taxa de 23750 FIT, ou apenas 42100 horas de uso antes de apresentar uma falha em um bit. Este fluxo foi calculado considerando o valor de 5 Nêutrons/cm^2/h dado por Ziegler para o nível do mar em Nova Iorque, EUA. Este valor está dentro de uma margem de erro próxima a 15%, quando comparando com dados experimentais obtidos por Claudio Federico em seu trabalho sobre medidas do fluxo de Nêutrons na região de São Paulo (Federico *et al.*, 2010). Na tabela 1 aparecem os valores de fluxo para algumas localidades. Nesta tabela, a diferença de fluxo entre Nova Iorque e São Paulo está relacionada com a Latitude destas localidades. Devido à inclinação do eixo de rotação da Terra, em Latitudes maiores o caminho através da atmosfera é mais curto. Assim, mais partículas conseguem atravessar.

Tabela 1- Fluxo de Nêutrons em diversas localidades

Localidade	Altitude (m)	Fluxo Nêutrons (cm^2/h)
Nova Iorque	0	5
Nova Iorque	10000	1453
São Paulo	0	3,3
São Paulo	10000	475
Itatiaia – RJ	2400	23*
Itatiaia – RJ	2400	20

* Fluxo medido (Federico *et al.*, 2010)

Fonte: Autor

Empregando os valores de fluxo de Nêutrons em vários locais e altitudes nota-se como um cálculo assim é importante na avaliação da confiabilidade e segurança de um equipamento eletrônico. Se uma placa eletrônica utilizar 8 circuitos integrados desta memória do exemplo anterior, na altitude de 10.000 m sobre São Paulo, a taxa de falhas resultaria em apenas 5260 horas antes de um bit ser alterado por causa de Nêutrons na altitude típica de aviões comerciais. Esta taxa fica ainda menor considerando todos os demais componentes empregados, além das memórias.

Importante notar que esta taxa de falhas se refere apenas aos efeitos resultantes da interação de partículas ionizantes com os dispositivos. Além desta existem outras relativas a defeitos de fabricação (que se tornam notáveis depois de algum tempo de uso de um componente), de operação (como temperatura ou limites de tensão de alimentação, por exemplo), dentre outras. A contribuição de todas as taxas de falhas resulta no cálculo da probabilidade de falha de um equipamento.

2.3 Efeitos da radiação ionizante em materiais semicondutores

O estudo da interação de partículas ionizantes com a matéria traz a fundamentação necessária para compreender como estes eventos podem causar falhas funcionais em circuitos eletrônicos. Estas falhas podem ser permanentes (irrecuperáveis) ou temporárias (*soft error*). Além disso, o modo como as partículas interagem com a matéria determina a intensidade e o tipo de efeito causado.

As fontes de partículas energéticas são extraterrestres, como raios cósmicos e solares; provenientes de materiais radioativos terrestres, como os presentes em algumas ligas metálicas empregadas como solda eletrônica (Gaillard, 2011); compostos empregados na fabricação de circuitos integrados, como o BPSG (*Borophosphosilicate glass*) (Wang e Agrawal, Vishwani D., 2008); ou mesmo ambientes inerentemente radioativos, como instalações nucleares.

2.4 Falhas permanentes

As partículas de alta energia podem causar danos permanentes nas estruturas dos materiais. Estes danos são cumulativos e de dois tipos: a Dose por Dano de Deslocamento (*Displacement Damage Dose*, DDD) e a Dose Ionizante Total (*Total Ionizing Dose*, TID).

Os danos por DDD (Xapsos *et al.*, 2017), também conhecido como TNID (*Total Non-Ionizing Dose*) (Trivedi e Mehta, 2016), ocorrem quando uma partícula pesada atinge o núcleo de um átomo de Silício e a energia é suficiente para mover o átomo de lugar. Este mecanismo cria o chamado defeito de Frenkel (Poivey, 2017)(Todd e Uznanski, 2015) que resultam em regiões vazias e alojamentos intersticiais, ou seja, um átomo deslocado aloja-se próximo de outros e com isso deixa vazia a região onde estava. Este defeito causa modificações nas características elétricas do material, como nas bandas de energia, que podem eventualmente causar falhas funcionais (Spieler, 1997). Cerca de 20 eV são suficientes para uma partícula deslocar um átomo de Silício de sua rede cristalina. Um Nêutron com energia de 1 MeV, atingindo um núcleo de Si, transfere de 60 a 70 keV ao rebater (*recoil*) e que pode resultar no deslocamento de 1000 outros átomos numa região de tamanho aproximado de 0,1µm (Spieler, 1997).

Embora os danos por DDD sejam permanentes seus efeitos podem efetivamente diminuir com o tempo. Isso acontece por causa da reação conhecida como recozimento (*annealing*) que acontece por até alguns minutos após o dano inicial (*short-term annealing*) e funciona basicamente por recombinação. Um recozimento de longa duração (*long-term annealing*) também ocorre mesmo em temperatura ambiente e pode se estender por vários meses (Srour, Marshall e Marshall, 2003). Este processo de recozimento permite que parte da rede cristalina contendo o deslocamento causado pela incidência da partícula possa ser

reordenada, reduzindo os efeitos iniciais. A recuperação, contudo, nunca é integral e por isso considera-se que estes danos sejam permanentes.

A medida NIEL é usada para indicar a quantidade de energia não ionizante que é transferida para um material. Sua unidade é a mesma de LET e pode ser calculada como (Srour, Marshall e Marshall, 2003):

$$\mathrm{NIEL} = (\mathrm{N/A})\,[\,\sigma_e T_e + \sigma_i T_i] \tag{2}$$

onde N é o número de Avogadro, A é o peso atômico do material em gramas, T_e e T_i são as energias efetivas de rebote elástica e inelástica. Já σ_e e σ_i são as secções de choque elástica e inelástica, respectivamente.

Em 1980 experimentos foram feitos com transistores bipolares irradiados com diversos tipos e intensidades de partículas para verificar os efeitos DDD (Srour, Marshall e Marshall, 2003). O resultado manteve o que foi obtido em um estudo anterior demonstrando que o ganho dos transistores decai linearmente proporcional ao fator de dano por deslocamento (Messenger e Spratt, 1958).

Os danos por deslocamento manifestam-se em três modos importantes (Spieler, 1997):

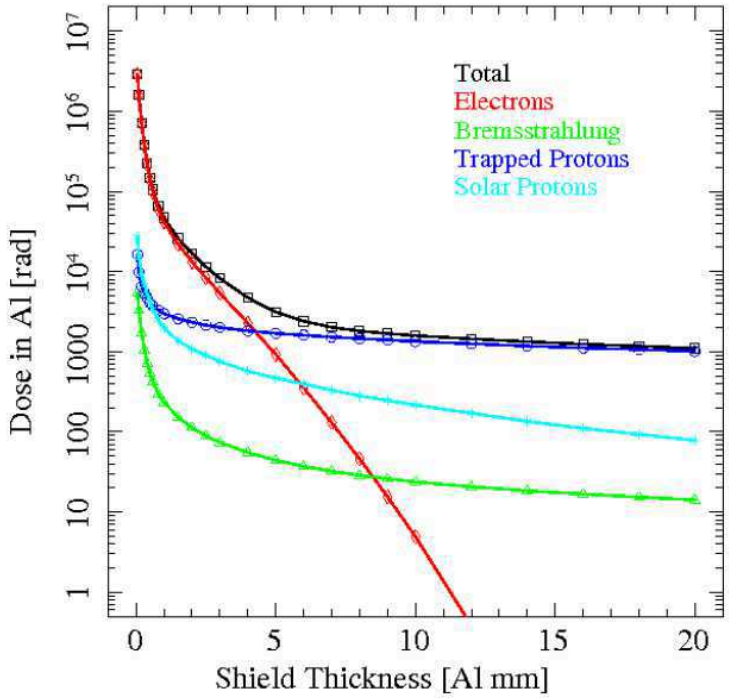
1) formação de novos níveis de energia na banda proibida (*band gap*) do semicondutor, facilitando a transição de elétrons entre a banda condutora e de valência. Este efeito aumenta a corrente elétrica em junções reversamente polarizadas (corrente de fuga).
2) facilitar o aparecimento de armadilhas de cargas que prendem cargas e as soltam depois de algum tempo.
3) mudança nas características de dopagem (densidade de aceitadores e doadores).

Já os danos por TID causam formação de armadilhas de lacunas (buracos) na camada de óxido das portas e de isolação entre transistores MOSFET. As lacunas presas no óxido de porta mudam as características da tensão de limiar (Vth) e aquelas presas no óxido isolante entre transistores faz aumentar a corrente de fuga dos dispositivos (Trivedi e Mehta, 2016). A corrente de fuga é considerada o efeito mais deletério por TID para aplicações científicas, aeroespaciais e militares (Faccio, 2000).

Em 1962 um teste nuclear realizado no Oceano Pacífico resultou num cinturão radioativo a 400 km de altitude formado por partículas geradas na explosão. Um satélite,

TELSTAR, acumulou TID 100 vezes maior do que o esperado e foi destruído. As consequências deste teste se estenderam e, num período de 7 meses, foram destruídos outros sete satélites por causa de TID (Baylakoglu e Hudaverdi, 2012). A energia de partículas cósmicas e solares é tão alta que o uso de blindagens metálicas não ajuda muito para proteger de TID como visto na Figura 1, onde apenas os elétrons são parados, considerando-se observações feitas pelo satélite RASAT localizado em órbita baixa (*Low Earth Orbit*, LEO).

Figura 1 - Efeito da espessura da blindagem de Alumínio contra radiação



Fonte: Extraído de (Baylakoglu e Hudaverdi, 2012)

O TID também afeta a sensibilidade do circuito a SET. Estudos mostram que a duração do pulso de corrente causado por um SET diminui proporcionalmente ao acumulo de TID (Balasubramanian *et al.*, 2008). Por outro lado, o TID aumenta a sensibilidade à SEU e interferências eletromagnéticas (EMI) (Villa *et al.*, 2017) (Koga *et al.*, 2009). O TID não é reversível, sendo considerado um parâmetro muito importante na seleção de componentes eletrônicos de uso espacial porque a manutenção para troca dos componentes atingidos não é normalmente possível. Contudo, alguma recuperação por *annealing* pode ocorrer, como observado em diversos estudos (Silveira *et al.*, 2015) (Kamarou, 2006).

Importante lembrar que outros efeitos causados por partículas energéticas como o SEB e o SEGR são também destrutivos, embora ocorrem imediatamente (não são cumulativos) no dispositivo atingido.

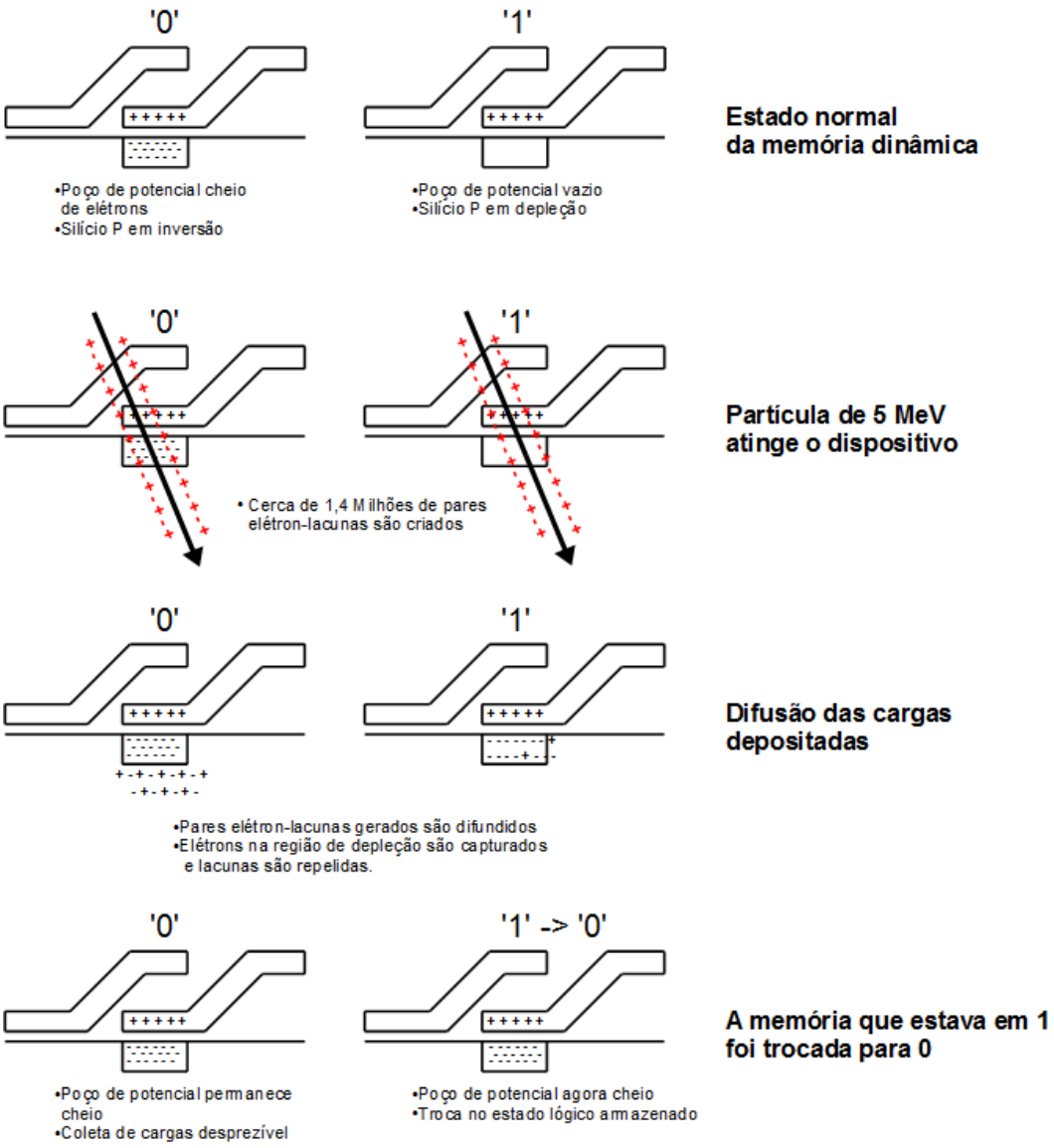
2.5 Fontes de radiação que causam falhas não permanentes

As três principais fontes de radiação que podem causar efeitos temporários nos semicondutores são:

1) Partículas alfa (núcleos de Hélio formados por dois prótons e dois nêutrons), emitidas quando o núcleo de um isótopo instável como Tório, Rádio ou Actínio, decai para um estado de energia mais baixo (Gaillard, 2011). Eles contêm energia cinética baixa entre 4 a 9 MeV. No ambiente terrestre, as principais fontes de partículas alfa são impurezas radioativas que ocorrem, por exemplo, nas esferas de solda empregadas em circuitos integrados de tecnologia BGA e flip-chip (Kumar, Agarwal e Jung, 2013). Este fenômeno ocorre tanto nas esferas de solda à base de chumbo quando nas livres de chumbo (*lead free*). Estas partículas alfa de baixa energia (também chamadas e ULA – *Ultra-Low Alpha*) causam efeitos significativos porque apesar de terem baixa energia são geradas por material localizado muito próximo ao silício do dispositivo (Wang e Agrawal, Vishwani D, 2008).

O efeito das partículas alfa em dispositivos semicondutores pode ser percebido nas ilustrações da Figura 2 que mostra como um elemento (bit) de uma memória dinâmica pode mudar do valor 1 para o 0 após a incidência de uma partícula com 5 MeV de energia (May e Woods, 1979).

Figura 2 – Alteração de um bit de memória



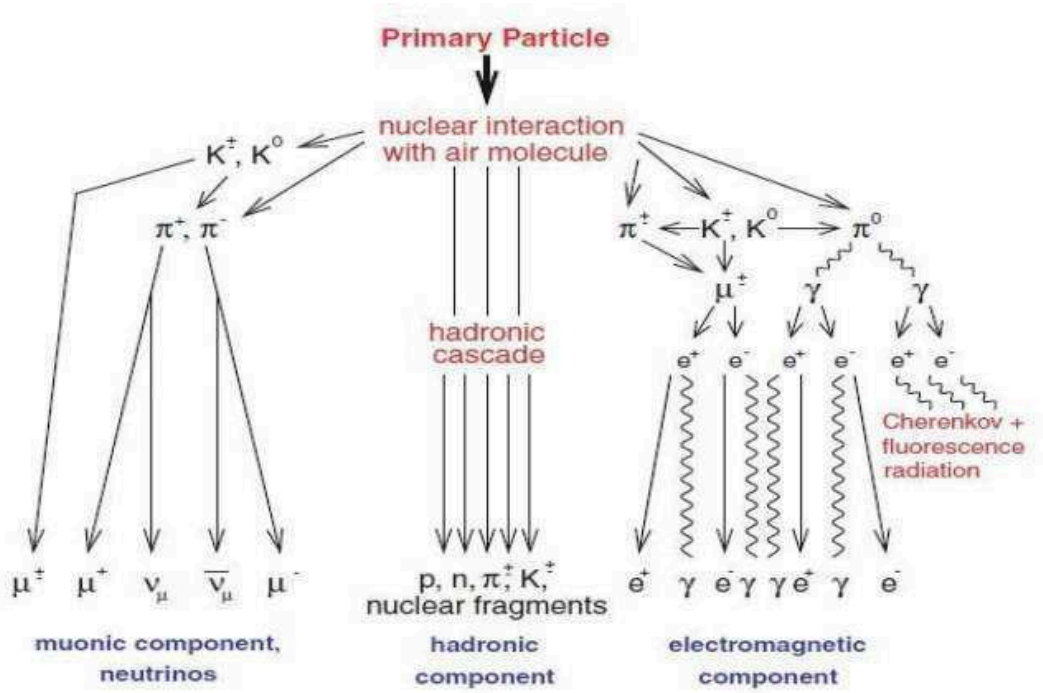
Fonte: Adaptado pelo autor de May e Woods, 1979, p. 4

O poço de potencial do elemento de memória (bit) contém aproximadamente 1 milhão de elétrons quando está registrando o valor 0 (May e Woods, 1979). Se o valor do elemento é 1 não existem muitos elétrons armazenados. Quando uma partícula Alfa atinge o dispositivo são gerados cerca de 1,4 milhões de pares elétron-lacuna numa profundidade de até alguns µm. Muitos elétrons gerados são coletados pelo campo elétrico do poço vazio e se a quantidade for suficiente a carga resultante representará o valor 1 para este elemento de memória.

2) Nêutrons de alta energia (1 MeV até 10GeV) (Gaillard, 2011) da radiação cósmica podem induzir erros em dispositivos semicondutores através de íons secundários produzidos

pela reação de nêutrons com núcleos de silício. Raios cósmicos que são de origem galáctica reagem com a atmosfera da Terra para produzir complexas cascatas de partículas secundárias (Figura 3) (Brancus e Rebel, 2015). Menos de 1% do fluxo primário atinge o nível do solo e as partículas predominantes incluem fótons (γ), múons (μ), nêutrons (N), prótons (p), elétrons (e-), pósitrons (e+), pions (π) e neutrinos (v) (Wang e Agrawal, Vishwani D, 2008). Uma vez que pions e múons são de curta duração e prótons e elétrons são atenuados pela interação eletrostática com a atmosfera, os nêutrons são as fontes de radiação cósmica mais prováveis para causar um SEE em semicondutores, localizados desde a altitude utilizada por aeronaves comerciais até no nível do mar. O fluxo de nêutrons é dependente da altitude e a densidade aumenta com a altitude. O espectro de energia das partículas geralmente não depende da altitude. Além disso, a atmosfera é importante atenuador do fluxo destas partículas (Gaillard, 2011).

Figura 3 - Cascata de partículas na atmosfera



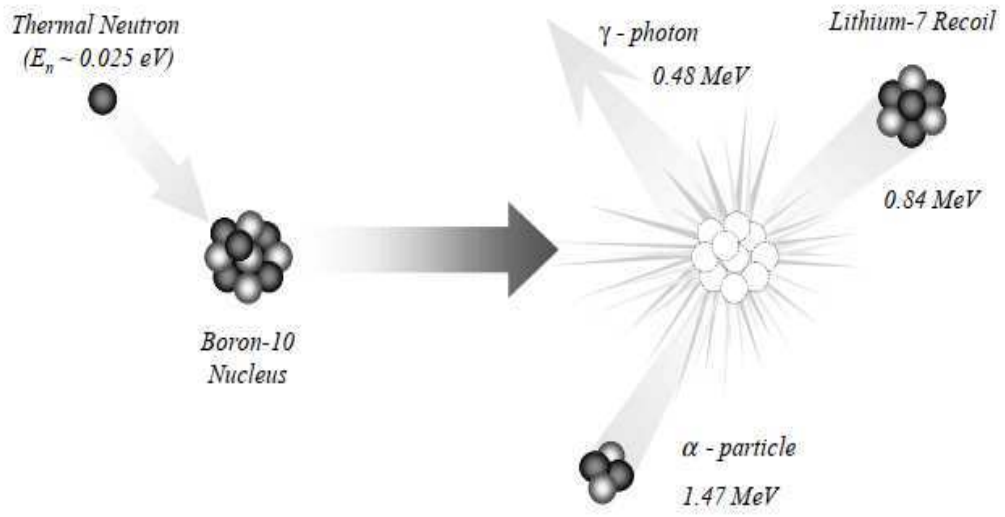
Fonte: Extraído de (Brancus e Rebel, 2015)

Os efeitos da interação direta de nêutrons com núcleos de silício produzem uma variedade de partículas secundárias, como prótons e alfas, que resultam em grande quantidade

de pares elétron-lacunas, ao atravessarem o material semicondutor e que efetivamente causam SEE.

3) A terceira fonte significativa de partículas ionizantes para dispositivos eletrônicos é a radiação induzida pelo nêutron de baixa energia com o isótopo Boro-10 (^{10}B é um dopante tipo-P comumente utilizado em circuitos integrados) (Gaillard, 2011). Especificamente, a camada dielétrica de BPSG (vidro de borofosfosilicato) que tem sido empregada para camadas isoladoras na fabricação de circuitos integrados. O esquema de reação é mostrado na Figura 4. Na reação do nêutron com o núcleo de Boro-10 este é transmutado para Boro-11 que é instável e decai emitindo um núcleo de Lítio-7 com uma cinética energia de 0,84 MeV (isso ocorre 94% das vezes e com 1,014 MeV em 6% das vezes). O fóton gama tem energia de 478 keV, enquanto a partícula alfa é emitida com uma energia de 1,47 MeV (1,78 Mev em 6% das vezes) (Baumann e Smith, 2001). Este mecanismo foi descoberto como sendo a fonte dominante de erros de software em memórias SRAM de tecnologia 0,25 e 0,18µ fabricada com BPSG (Wang e Agrawal, Vishwani D, 2008). O avanço tecnológico tem permitido reduzir as impurezas nos materiais empregados nos circuitos integrados e este mecanismo de ação tem sido menos frequente.

Figura 4 - Reação nuclear no Boro-10 atingido por um Nêutron



Fonte: Extraído de (Baumann e Smith, 2001)

2.6 Transferência Linear de Energia

A energia depositada por uma partícula no material do dispositivo atingido é denominada Transferência Linear de Energia ou LET (*Linear Energy Transfer*), com interpretação similar e valores idênticos ao parâmetro conhecido em física como *Stopping Power*. Sua unidade de medida é MeV·cm²/mg e envolve a área do material atingido, sua massa e a energia da partícula. A imunidade de um dispositivo é medida pelo parâmetro LET_{th} (Holbert, 2006), determinando a energia mínima para que uma partícula possa causar um SEE quando existe uma fluência de 10^7 íons/cm^2. Como visto anteriormente, medida da área atingida por partículas energéticas é chamada de secção (*cross section)*.

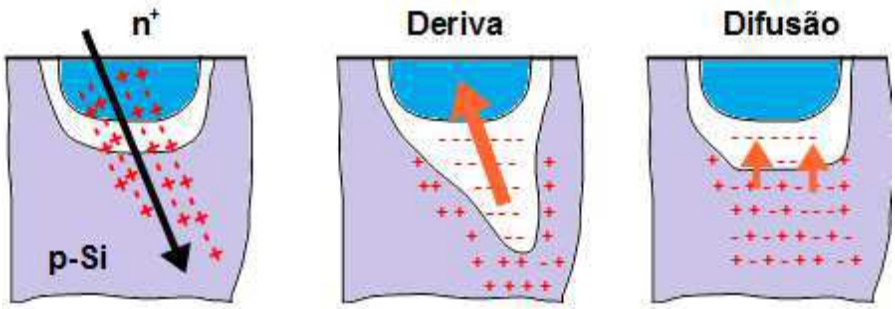
Um dispositivo totalmente imune à SEE possui LET_{th} > 100 MeV·cm²/mg (LaBel, 1996). O LET_{th} usualmente reduz conforme o dispositivo acumula grandes doses de radiação (*Total Ionizing Dose*, TID) (Trainor, 1994).

2.7 Efeitos da interação das partículas energéticas em transistores

Em circuitos CMOS, os transistores desligados atingidos por uma partícula energética são mais sensíveis à SEU ou SET causado por partícula ionizante com LET suficientemente alto, a partir de aproximadamente 20 MeV·cm²/mg. Quando essas partículas atingem o silício, portadores minoritários são criados e caso sejam coletados pelas regiões de difusão da fonte ou dreno, uma mudança no valor da tensão desses pontos pode ocorrer. Uma partícula pode induzir um evento quando atinge a região do canal de um transistor NMOS desligado ou a região de dreno de um transistor PMOS desligado (Wang e Agrawal, Vishwani D, 2008).

Conceitualmente, quando a carga injetada pela radiação em um ponto sensível excede a carga crítica (Q_{crit}), um evento único é gerado na região afetada.

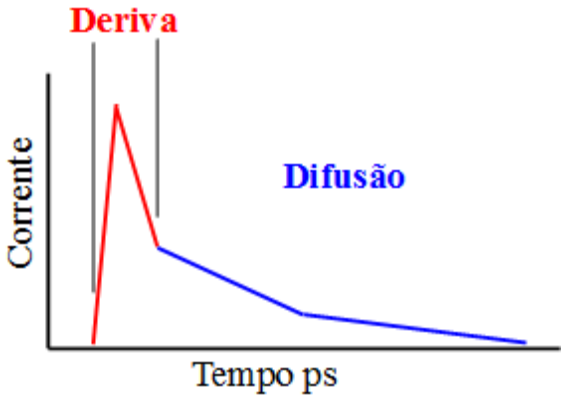
Figura 5 - Interação de uma partícula de alta energia



Fonte: Autor

Na Figura 5, acima, uma partícula de alta energia atinge uma junção PN. Ao longo do caminho percorrido, a partícula produz uma distribuição radial densa de pares de elétrons-lacunas como ilustrado, formando um funil na região de depleção, onde os portadores são rapidamente coletados pelo campo elétrico, compensando assim a carga armazenada na junção. O funil temporariamente formado, que alonga a região de depleção ao longo da trajetória da partícula, tem o efeito de aumentar ainda mais a coleta de cargas por deriva. Depois disso a coleta de carga continuará através da difusão.

Figura 6 - Corrente elétrica na coleta de cargas



Fonte: Autor

A fase de coleta por deriva normalmente dura dezenas de picosegundos e conforme o funil colapsa o processo é dominado pela difusão das cargas (Figura 6) que pode durar até alguns nanosegundos. A carga transitória coletada (Q_{coll}) pode produzir um pulso de corrente num transistor desligado nos mesmos níveis que corresponderia à corrente esperada em um

transistor ligado. Esta corrente efetivamente causa um SET que pode se tornar um SEU caso atinja um circuito de memória.

A maior parte da coleta de cargas ocorre dentro de 2 a 3 µm da região de junção para tecnologias CMOS (Wang e Agrawal, Vishwani D, 2008). As constantes de tempo dependem fortemente do tipo de íon, da sua energia inicial e das propriedades específicas da tecnologia. A carga coletada (Q_{coll}) é uma função da energia e trajetória da partícula ionizante, estrutura e dopagem do substrato de silício e do campo elétrico local.

Um modelo analítico aproximado da corrente transiente induzida para coleta de cargas é representado na Equação 3:

$$I(t) = \frac{Q_{coll}}{\tau\alpha - \tau\beta}\left(e^{-t/\tau\alpha} - e^{-t/\tau\beta}\right) \tag{3}$$

onde Qcoll é a carga coletada na região sensível (em femto coulomb, fC), τ_α é o tempo de coleta que dependente do processo de fabricação do dispositivo e τ_β é o tempo para ocorrer o funil ionizado que é relativamente independente da tecnologia. Valores típicos são aproximadamente $1{,}64 \times 10^{-10}$s para τ_α e 5×10^{-11}s para τ_β (Wang e Agrawal, Vishwani D, 2008). De fato, diversos estudos mostram que a duração total de um SET não chega a 2 nanosegundos (Wirth, Kastensmidt e Ribeiro, 2008).

O valor de Qcoll depende da profundidade em que ocorre a coleta de cargas e da energia transferida (LET). Tipicamente uma partícula ionizante deposita 10,8 fC (femto coulomb) a cada 1µm. Deste modo chega-se a Equação 4 (Wang e Agrawal, Vishwani D, 2008):

$$Q_{coll} = 10{,}8 * L * LET \tag{4}$$

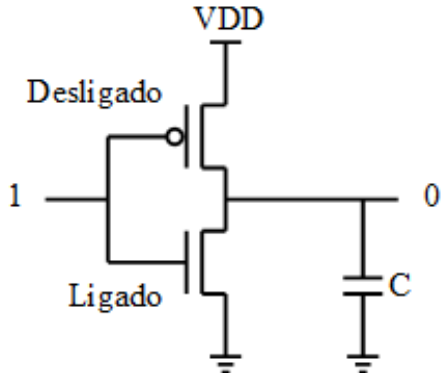
onde L é a profundidade de coleta de cargas em µm. Outro modo de calcular a corrente produzida pela partícula energética é considerando a mobilidade das cargas, o campo elétrico (dependente da tensão de operação do dispositivo) e a energia transferida (LET) (Truyen *et al.*, 2007):

$$I = \langle \mu E_{Field} \rangle \, LET \tag{5}$$

onde μ e E_{Field} representam a média da mobilidade do elétron e campo elétrico, respectivamente. Embora a mobilidade dos elétrons tenha forte dependência da temperatura para a corrente elétrica resultante esta não muda significativamente a sensibilidade à SEU porque a tensão de limiar dos dispositivos acompanha inversamente a temperatura, compensando o efeito (Truyen *et al.*, 2007).

A carga mínima necessária para causar um SEU ou SET é denominada carga crítica ou Q_{crit} e pode ser calculada empregando a Equação 6 (Wang e Agrawal, Vishwani D, 2008). Uma partícula pode induzir um evento único ao atingir a região do canal de um transistor NMOS desligado, ou a região de dreno de um transistor PMOS desligado. O transistor sensível é geralmente aquele que está desligado porque no que está ligado a corrente das cargas coletadas não fará efeito contrário ao já existente no circuito. O caso do PMOS desligado será considerado como exemplo ilustrativo, descrito a seguir e esquematizado no diagrama das Figuras 7 e 8.

Figura 7 - Esquema de um inversor



Fonte: Autor

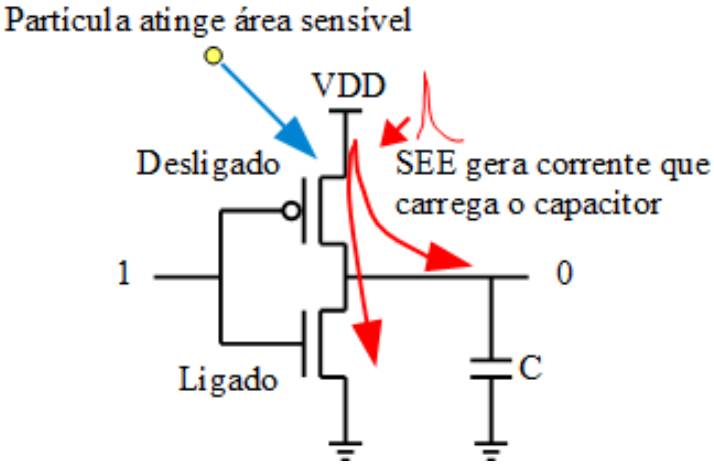
A carga crítica Q_{crit} depende da carga total coletada na região sensível, da duração necessária para o pulso de corrente resultante e da tensão de alimentação do dispositivo. Chama-se de tempo de comutação (t_{th}) o intervalo entre a colisão da partícula e a tensão na saída do dispositivo chegar no nível limite em que o transistor seguinte no circuito possa reagir ao impulso. Q_{crit} pode ser calculado integrando a corrente que flui na área sensível após a partícula atingir o dispositivo, como mostrado na Equação 6 (Detcheverry *et al.*, 1997).

$$Q_{crit} = \int_0^{t_{th}} I_{dreno}(t)dt \tag{6}$$

A corrente $I_{dreno}(t)$ no transistor PMOS é a soma da corrente que pode fluir no transistor NMOS, que está ligado, com a corrente necessária para carregar o capacitor de saída (Mohanram, 2005). O limite de corrente no NMOS é dado pela resistência de canal deste transistor. Quando o PMOS é atingido por uma partícula, a capacitância efetiva na saída do circuito (C) será carregada com Q_{crit}. Neste momento a largura do pulso de tensão que aparece dependerá do valor da capacitância e da constante de tempo RC existente no caminho de descarga através do transistor NMOS, que continua ligado (Figura 8). A tensão resultante da ionização é dada por Q_{crit}/C.

Por exemplo, num exemplo de circuito, quando a capacitância de carga de saída é 100 fF e a carga coletada é 0,65 pC, a amplitude do pulso de tensão é 0,65 pC / 100 fF = 0,65 V. Observa-se que para uma mesma carga coletada se a capacitância for menor a tensão resultante será maior (Wang e Agrawal, Vishwani D, 2008). Portanto, como a capacitância de carga é dependente da tecnologia, que tende a diminuir em função da redução das dimensões (tecnologias nanométricas e consequente aumento de velocidade dos circuitos), a tensão resultante devida ao SEE tende a aumentar nestas tecnologias, facilitando o acionamento dos transístores seguintes, conectados no circuito, tornando estes dispositivos mais susceptíveis à radiação.

Figura 8 - Caminho da corrente elétrica induzida por uma partícula

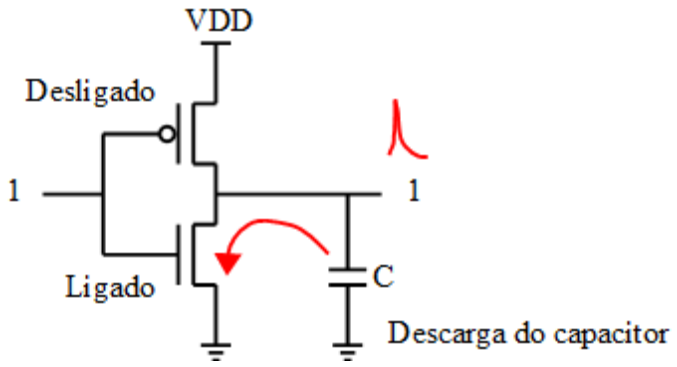


Fonte: Autor

O processo de carga e descarga pode ser modelado por um simples circuito RC e esta abordagem tem sido empregada para facilitar a simulação de SEE em circuitos eletrônicos (Mohanram, 2005). A Figura 9 mostra a descarga através do transistor NMOS. Durante um

breve momento a saída do circuito apresenta tensão alta o suficiente para ser interpretada como nível lógico 1.

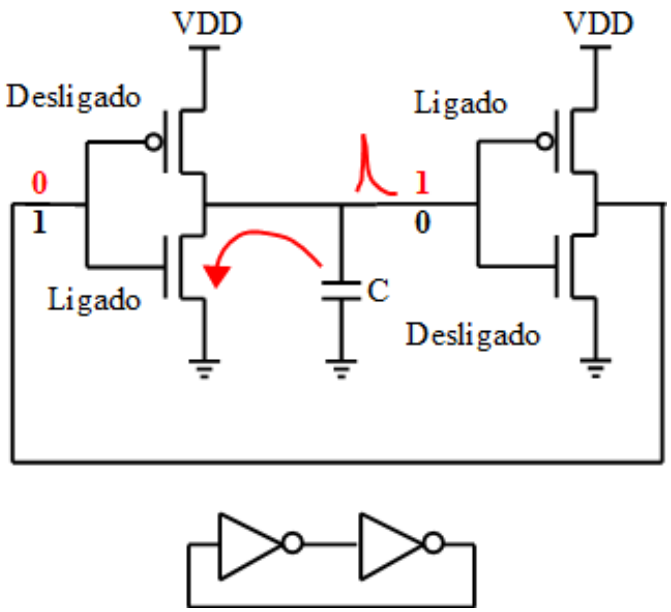
Figura 9 - Descarga do capacitor de saída



Fonte: Autor

Em um circuito de memória típico, o esquema é representado por duas portas inversoras, como das figuras, ligadas em circuito-fechado (Figura 10). Uma ocorrência de um evento único, com esta tensão alta, vai fazer o inversor seguinte trocar o valor de sua saída de 1 para 0. Este 0 vai então estar presente na entrada do primeiro inversor fazendo sua saída manter o valor 1 estimulado pela partícula ionizante e assim causando o que se chama, em Inglês, de *bit-flip* (troca de valor de um bit, de 0 para 1 ou vice-versa).

Figura 10 - Célula de memória



Fonte: Autor

2.8 Técnicas para Reduzir a Sensibilidade e Efeitos causados por SEE

A quantidade de transistores em memórias e processadores digitais produzidos atualmente está em constante aumento. Com suas dimensões cada vez menores, tais circuitos tornam-se mais susceptíveis a efeitos causados por partículas de alta energia. Muitas técnicas têm sido desenvolvidas para mitigar estes efeitos, explorando tanto o desenho de construção dos dispositivos, suas propriedades físico-químicas como também arranjos de circuitos que consigam recuperar informações alteradas por SEE.

O conjunto destas técnicas é chamado de *Radiation Hardening by Design* ou RHBD. Além da possibilidade de escolha dos materiais estas técnicas são aplicadas de dois modos: por modificação do leiaute ou por modificação do desenho do circuito.

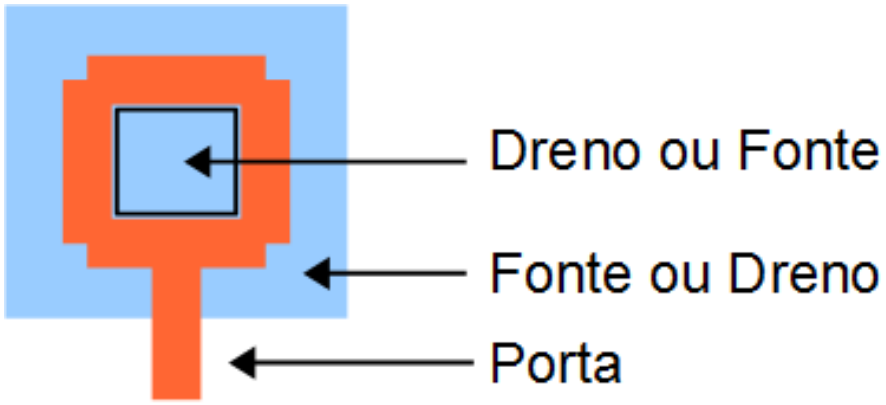
2.9 RHBD por modificação do leiaute

Para mitigar os efeitos causados por SEE, em especial TID e SEL, diversas técnicas tem sido estudadas, em particular o leiaute de transistor fechado, o anel de proteção (Trivedi e Mehta, 2016) e modificações no dimensionamento (Cazeaux, Oma e Nw, 2005).

2.9.1 ELT

O leiaute de transistor fechado (*Enclosed Layout Transistor* ou *Edgeless Layout Transistor*, ELT) auxilia na prevenção de corrente de fuga entre Fonte e Dreno (Anelli *et al.*, 2000). Este tipo de leiaute é também chamado de EG (*Enclosed Gate*) (Bucher *et al.*, 2018). Nesta técnica a Porta do transistor é desenhada formando um anel em torno da Fonte ou do Dreno, como mostrado na figura 11.

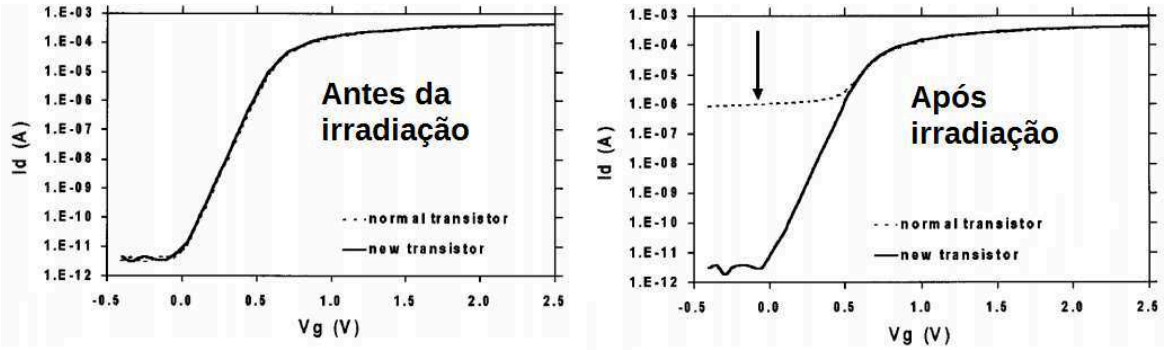
Figura 11 - Leiaute de um transistor fechado



Fonte: Autor

O ELT, ou EG, é um desenho muito eficaz para proteger um transistor contra doses de radiação muito elevadas (Faccio, 2005) (Bucher *et al.*, 2018). A Figura 12 mostra os efeitos da radiação em um transistor com leiaute convencional comparado com um transistor com leiaute ELT (Snoeys, Gutierrez e Anelli, 2002). Nesta figura a curva pontilhada indica os dados de um transistor normal e a curva contínua (*new transistor*) indica um transistor ELT. Contudo, a dificuldade em modelar o ELT (relação dimensional W/L) e sua inerente assimetria (áreas de Dreno e Fonte) resultam em não existir grandes esforços para desenvolvimento de bibliotecas com esta tecnologia (Souza, Silveira e Gimenez, 2014). Poucas bibliotecas foram desenvolvidas e são tipicamente empregadas em aplicações especiais como em sistemas integrados para o CERN e equipamentos para uso espacial (Faccio, 2005).

Figura 12 - Modelo ELT sob radiação



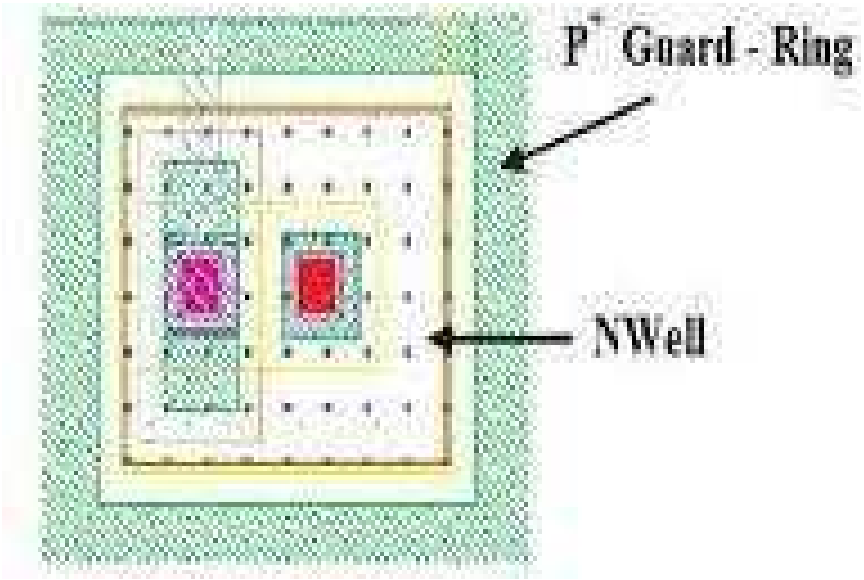
Fonte: Adaptado de (Snoeys, Gutierrez e Anelli, 2002)

2.9.2 Guard Ring

O anel de proteção (*Guard Ring*) reduz a corrente de fuga entre transistores e o ganho dos transistores parasitas PNPN intrínsecos à tecnologia CMOS. Ao reduzir o ganho destes transistores parasitários o risco de SEL é também reduzido pois exigem energia maior para iniciar a condução. Foi demonstrado que dispositivos empregando anéis de proteção apresentam alta tolerância para SEL com LET_{th} > 90 MeV·cm²/mg (Trivedi e Mehta, 2016). Exemplo desta técnica está na figura 13. O uso do anel de proteção em conjunto com o leiaute ELT proporciona grande eficácia na redução da sensibilidade à SEL e os efeitos cumulativos por TID (Anelli *et al.*, 2000).

O anel feito com P^+ num substrato P cria uma efetiva barreira impedindo a formação de inversão nesta área por causa de cargas depositadas por partículas ionizantes.

Figura 13 - Anel de proteção



Fonte: Extraído de (Trivedi e Mehta, 2016)

2.9.3 Ajuste dimensional

Uma forma de reduzir a sensibilidade às falhas como SEU e SET é aumentar o valor necessário para Q_{crit}, exigindo maior quantidade de energia transferida pela partícula incidente

para causar uma falha (Cazeaux, Oma e Nw, 2005). Isso pode ser feito redimensionando a largura do canal que resulta em aumento das dimensões da porta lógica (*gate*). Dobrando a largura do canal foi observado um aumento de 58% no valor de Q_{crit}.

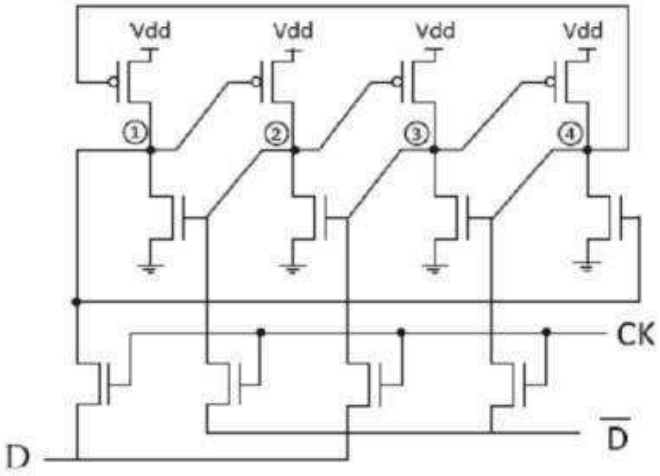
O balanço na geometria entre transistores NMOS e PMOS em um inversor também influencia a resposta a um SET. Em alguns casos a dominância de um ou de outro transistor, dependendo da tecnologia utilizada, pode filtrar ou reforçar um SET (Rezgui, Mccollum e Won, 2010).

2.10 Arquiteturas Digitais

Em outra abordagem tecnológica, dentro das técnicas para prover tolerância às falhas por SEE, empregam-se arquiteturas digitais que permitem a correção ou filtragem do efeito causado pela partícula incidente. Nos computadores, os elementos mais críticos, no caso de ocorrências de SEE, são células de memória e registradores (*flip-flops*). Extenso trabalho tem sido desenvolvido para aperfeiçoar o desenho de memórias. Uma síntese disso foi publicado na pesquisa de Trivedi sobre o estado da arte em técnicas para reforçar circuitos eletrônicos de aplicação espacial (Trivedi e Mehta, 2016).

Em geral emprega-se redundância de transistores de forma que a injeção de corrente em um deles, por efeito da incidência de uma partícula de alta energia, seja compensada pelo estado dos demais transistores do circuito (Chertkow *et al.*, 2015). O exemplo mais conhecido é a memória DICE (acrônimo para *Dual Inter Clocked storage Cell*) mostrada na Figura 14. Nesta arquitetura dois registradores são interligados de forma que os transistores NMOS e PMOS não ficam ligados nas mesmas entradas (Trivedi e Mehta, 2016).

Figura 14 - Esquema de uma memória DICE



Fonte: Extraído de (Trivedi e Mehta, 2016)

O desenvolvimento de desenhos de circuito resistentes à SEE acontece desde 1988 quando o *Rockett Memory Cell* foi apresentado (Rockett, 1988). O objetivo das pesquisas relacionadas com células de memória tem sido reduzir o número de transistores necessários para garantir a integridade do dado armazenado mesmo na ocorrência de um SEE. Nesta linha foram desenvolvidos circuitos com duplo transistor como na célula de memória SHIELD (*SEU Hardening Incorporating Extreme Low power bitcell Design*) (Chertkow *et al.*, 2015) ou com intertravamento triplo (Li *et al.*, 2014).

Os registradores que armazenam bits de informação como parte de circuitos complexos, como nos núcleos de computadores (CPU), também tiveram suas arquiteturas melhoradas para recuperarem o valor de um bit corretamente caso sofra um distúrbio por SEU ou SET. São exemplos disso os desenhos de registradores chamados SETTOFF (Lin e Zwolinski, 2012) e RAZOR (Ernst *et al.*, 2003).

Algumas técnicas de redundância também são aplicadas em *flip-flops* para reduzir as taxas de falhas, incluindo redundância lógica e temporal.

2.11 Redundância

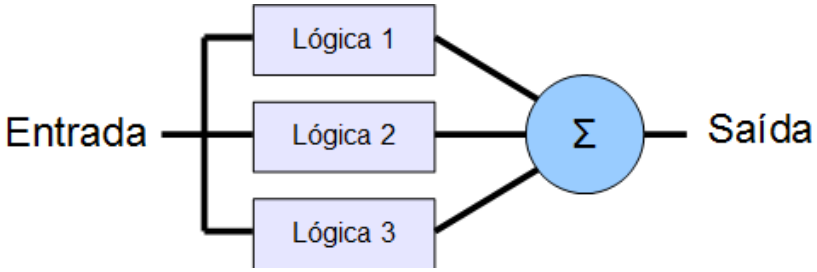
O emprego de circuitos redundantes é uma técnica de arquitetura digital empregada desde o desenvolvimento de computadores de controle de foguetes como o Saturno V (Wakerly,

1975) e tem sido explorada tanto no emprego de hardware redundante quanto em software (Kim e Shanbhag, 2010).

Denomina-se Redundância Dinâmica aquela em que uma falha é detectada e posteriormente corrigida pelo próprio circuito, enquanto que a Redundância Modular caracteriza-se por replicar os circuitos críticos e, através de uma comparação dos resultados destes circuitos, utilizar apenas a informação considerada correta (McCluskey, Wakerly e Ogus, 1975). A redundância N-modular (NMR), com uma quantidade N de módulos idênticos, permite um aumento da confiabilidade de um sistema, porém de forma limitada a eventos intermitentes e ainda com o acréscimo de custo inerente ao aumento de área de Silício e energia consumida (Joshi, Pradhan e Mohanty, 2010). A confiabilidade é a probabilidade de um sistema funcionar corretamente durante um certo período de tempo. Neste caso, a redundância modular resulta em aumento da confiabilidade porque mesmo que um dos módulos exiba uma falha o funcionamento dos demais garante que o sistema, como um todo, continue funcionando corretamente. A limitação a eventos intermitentes significa que a NMR pode resolver problemas transitórios (como os *soft-errors* causados por SEU ou SET) mas não resolve problemas constantes como as falhas resultantes de defeitos de fabricação ou componentes danificados (Joshi, Pradhan e Mohanty, 2010) (Koren e Su, 1979).

O modelo de redundância NMR mais comumente empregado é o TMR (*Tripple Modular Redundancy*), exemplificado na figura 15. A operação esperada para o circuito é replicada três vezes e uma votação majoritária é realizada para garantir que o resultado seja correto caso pelo menos duas das três cópias do circuito apresente o mesmo resultado. A votação, quando se trata de um sinal digital simples, é efetivamente a soma dos três resultados. Se o resultado das lógicas fosse um valor (barramento digital de múltiplos bits) o circuito de votação teria que comparar os três valores e escolher o que fosse igual em pelo menos duas das lógicas.

Figura 15 - Circuito típico para um TMR



Fonte: Autor

Quando empregada em sinais digitais simples (um bit) ou barramentos de dados (múltiplos bits) a técnica NMR funciona bem. Em sistemas complexos ou envolvendo computação distribuída a NMR encontra dificuldades (Hammett, Charles e Draper, 2001).

Esta dificuldade da NMR em casos de informação distribuída pode ser exemplificada com o clássico problema dos Generais Bizantinos (Lamport, Shostak e Pease, 1982). Neste problema alguns generais comandam tropas de ataque a uma cidade, vindo cada tropa por um dos lados. A troca de mensagens entre eles é feita por mensageiros que precisam passar próximo da cidade, onde podem ser capturados. Como a comunicação entre os generais não consegue ser confirmada tanto no seu conteúdo quanto se realmente aconteceu (no caso de um mensageiro ser capturado) fica impossível se coordenarem. Além disso, não é sabido se todos generais são leais às ordens do Imperador. A falha não está na integridade da informação, pois pode estar assinada por um dos generais junto com seu selo oficial, mas sim no sentido de seu conteúdo. Nesta alusão, com um sistema NMR com três cópias de dados (TMR), pode ocorrer uma entrada alterada por causa de um SEE no momento da leitura dos dados, ficando aparentemente válida para mais de um dos circuitos, mas não para todos. Com isso a decisão do circuito de votação fica comprometida apesar de sua lógica estar correta.

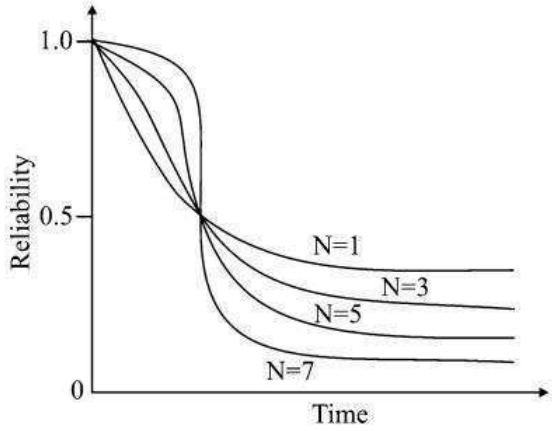
Já a verificação da integridade dos dados em um barramento digital pode ser feita por paridade, onde uma falha em quantidades ímpares de bits pode ser detectada. Porém, uma falha em quantidades pares de bits pode passar despercebida (Koopman e Driscoll, 2012).

O intervalo de tempo mínimo em que um sistema deve funcionar sem falhas irrecuperáveis é chamado “tempo de missão” (Koren e Su, 1979). A confiabilidade de um sistema empregando NMR depende do tempo de missão e da probabilidade de falhas neste tempo, considerando inclusive o aumento da quantidade de dispositivos resultante da redundância implementada. Quanto mais dispositivos em um circuito maior a chance de ocorrer uma falha (Torell e Avelar, 2004).

Uma redundância com 7 cópias, mostrado no gráfico da figura 16, por exemplo, apresenta uma excelente confiabilidade inicialmente mas depois de um certo tempo aproxima-se de zero (Joshi, Pradhan e Mohanty, 2010). O tempo de missão, portanto, deve ser restrito ao intervalo em que a confiabilidade permaneça muito próxima de 1, mas deve ser considerado que depois disso a probabilidade de falha aumentará progressivamente em relação à quantidade de módulos redundantes implementados.

O aumento de confiabilidade empregando NMR é, portanto, um efeito temporário.

Figura 16 - Confiabilidade de circuitos NMR

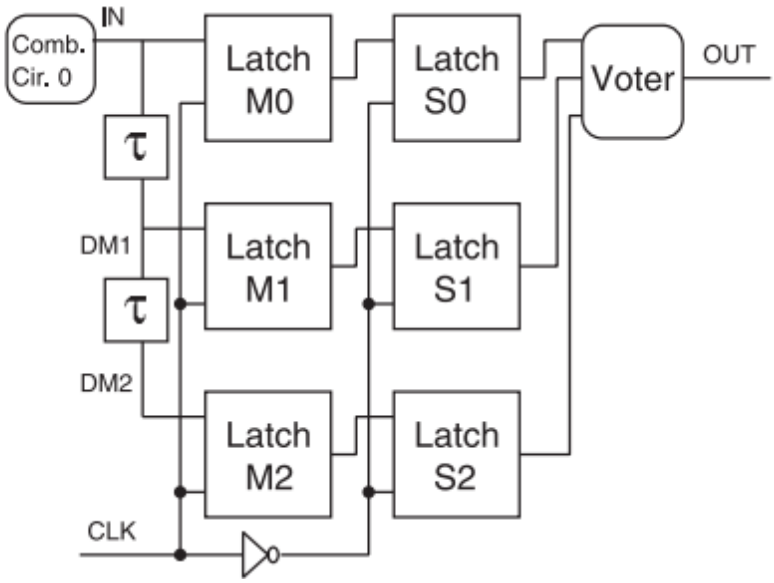


Fonte: Extraído de (Joshi, Pradhan e Mohanty, 2010)

A redundância pode ser explorada também temporalmente, aproveitando a curta duração dos transientes causados por SET para estabelecer que algumas cópias de um dado estejam válidas quando deslocadas no tempo em relação a uma cópia que tenha sido alterada pelo evento único (Anghel, Alexandrescu e Nicolaidis, 2000). A redundância temporal e modular para flip-flops, resultando no chamado DTMR (*Delayed Tripple Modular Redundancy*), permite obter confiabilidade bastante alta embora tenha a penalidade de consumir muita área de circuito (Figura 17). A figura mostra circuitos de atraso (τ) que transportam o valor da entrada (IN) para as demais cópias dos circuitos de memória (*Latch* M0, M1 e M2) em tempos diferentes. Assim, se ocorre um SET na entrada IN no mesmo instante do *clock* CLK ele pode ser armazenado em M0 mas não será armazenado em M1 e M2.

Alguns estudos exploram a combinação de atrasos e comparação de valores para reduzir a quantidade de circuito necessária sem penalizar muito a velocidade (Furuta, Kobayashi e Onodera, 2010).

Figura 17 - Circuito de um DTMR

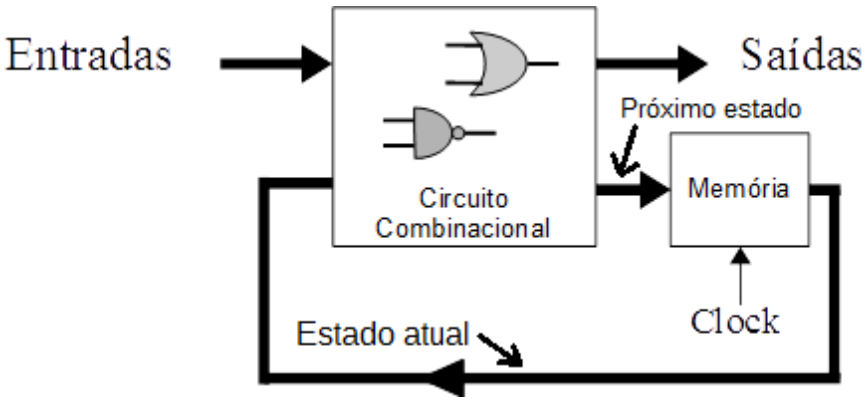


Fonte: Extraído de (Furuta, Kobayashi e Onodera, 2010)

2.12 Máquina de Estados Finita

Máquina de Estados Finita (*Finite State Machine*, FSM) é um circuito sequencial, ou seja, possui uma memória que registra o que ocorreu num instante anterior no tempo e que é usada no instante atual para determinar o que todo o circuito deve fazer. Para isso, um circuito lógico combina os sinais de entrada da FSM, com o que estava memorizado anteriormente, para determinar as saídas e o que deve ser memorizado no instante atual. No próximo instante os dados que haviam sido memorizados e os sinais de entrada serão tratados e novamente registrados, tal qual numa execução sequencial (Figura 18). Cada instante de uma FSM é chamado de estado. As FSMs são empregadas em qualquer processo que exija temporização.

Figura 18 - Máquina de Estados Finitos



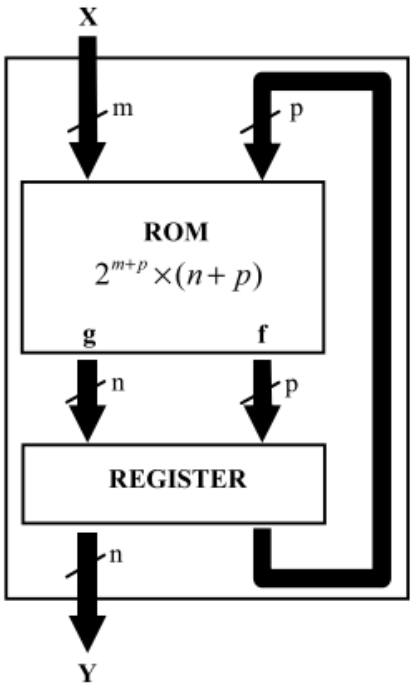
Fonte: Autor

As máquinas de estado são classificadas em dois tipos: Mealy e Moore. Uma FSM do tipo Mealy tem a saída dependente tanto do valor atual das entradas como do valor do estado atual. Esta característica torna as máquinas tipo Mealy inadequadas para aplicações críticas pois as saídas não são estáveis durante cada estado da máquina. Uma FSM do tipo Moore tem as saídas dependentes apenas do estado atual (Vaidya e Anupam, 2013) e o próximo estado é resultado da combinação do estado atual com os sinais de entrada.

A maneira de codificar cada estado pode ser implementada de diversas formas incluindo numeração binária, código Gray e *one-hot* (onde apenas 1 bit de um barramento fica ligado). O uso de numeração binária geralmente consome menos registradores (*flip-flops*) para armazenar o número do estado. Empregando 3 bits é possível identificar até 8 estados diferentes. Porém, a identificação binária é mais susceptível a falhas e temporizações (atrasos na propagação dos sinais resultaria na decodificação de um estado intermediário enquanto todos os bits não estiverem estáveis). Já o tipo *one-hot* emprega um único sinal para determinar qual o estado ativo, exigindo mais registradores (um para cada estado possível). O código Gray é uma representação binária em que a mudança de um estado para outro, quando em sequência, implica em mudar apenas um bit de um número binário (muito útil para reduzir consumo de energia porque ela depende da quantidade de sinais sendo comutados).

Além da FSM convencional existe a PSM (*Programmable State Machine*). Uma PSM emprega uma memória para armazenar todos os estados e todas as saídas possíveis. Portanto, em cada endereço da memória os dados guardados incluem tanto o endereço para o próximo estado quanto as informações que devem aparecer nas saídas durante o estado atual (Figura 19). Este tipo de arquitetura de máquinas de estado é considerado um método que melhora a confiabilidade do sistema (Frigerio e Salice, 2007).

Figura 19 - Máquina de Estados Programável



Fonte: Extraído de (Garcia-Vargas *et al.*, 2007)

O tipo de memória, tipicamente uma ROM (*Read Only Memory* – memória apenas de leitura), sugere que a PSM não seja programável por utilizar uma memória que não pode ser modificada. Contudo, uma ROM tem seu conteúdo criado em algum momento e de acordo com critérios da pessoa que projetou o circuito. Como este conteúdo é independente do circuito propriamente, ou seja, a funcionalidade da PSM pode ser diferente sem que o desenho do circuito dela seja alterado, então esta máquina de estados pode ser considerada programável. Mesmo que esta programação ocorra no nível de estabelecer os zeros e uns da memória durante o desenho das interconexões dos transistores no circuito integrado.

A ocorrência de SEE em máquinas de estados finitos pode causar falhas críticas. O circuito da central de processamento (CPU), em computadores e microprocessadores, é essencialmente uma máquina de estados que executa sequencialmente as operações de leitura e escrita de dados, cálculos e tratamentos de entradas e saídas. Em FPGAs (*Field Programmable Gate Arrays*) as FSM são especialmente susceptíveis porque como um SEE pode causar um distúrbio na memória que configura o circuito do FPGA então a falha, ocorrendo aleatoriamente, pode ser tal que o circuito da FSM seja modificado e, portanto, seu funcionamento é alterado mesmo que continue lidando sempre com dados e sinais confiáveis (Nidhin *et al.*, 2017).

2.13 Máquina de Estados Finita com maior confiabilidade

De acordo com alguns estudos 49% dos erros causados por SET ou SEU ocorrem em circuitos sequenciais (Kim *et al.*, 2005). Naturalmente, os esforços para desenvolver máquinas de estado tolerantes a falhas foram bastante focados nos registradores que armazenam os estados (El-maleh e Al-qahtani, 2014). Um dos trabalhos propõe a duplicação com auto verificação da codificação *one-hot* (Cassel e Kastensmidt, 2006). Este método garante que nenhum erro simples (SET, SEU) possa fazer a máquina ir para um estado incorreto, porém pode permitir um resultado incorreto no processo da FSM e implica na duplicação de todos os registradores de estado, ocupando bastante área de Silício. Já num outro trabalho emprega-se NMR para aumentar a tolerância a falhas, incluindo N circuitos de votação (Hadjicostis e Verghese, 2005) sem, contudo, evitar erros de resultado no processo.

Uma outra abordagem é armazenar a passagem por estados confiáveis e retornar a máquina para um estado confiável quando ocorre uma falha por SEE. Esta técnica é chamada *Rollback Recovery* e um estudo mostrou ter boa capacidade de recuperação de falhas múltiplas (Huang e Liang, 2009). Neste estudo feito por Zhengfeng Huang, alguns diferentes circuitos FSM foram testados na altitude de 60.000 pés e num dos casos o uso de *rollback recove*ry resultou em 7,4 vezes mais confiabilidade (num tipo de máquina de estados obteve FIT=0,308 para o circuito original e 0,042 para o circuito empregando *rollback recovery*). Este aumento de confiabilidade, contudo, ficou dependente do tamanho e complexidade da máquina de estados.

O uso da tecnologia TSC (*Totally Self-Checking*) apresenta outra forma de construir máquinas de estado finitas com maior confiabilidade (Zeng, Saxena e McCluskey, 1999). Nesta abordagem os circuitos da FSM são duplicados (em alguns casos parcialmente, com uso de reaproveitamento de circuitos lógicos) e cálculo de paridade é empregado para determinar se os resultados em cada estado estão corretos (Djordjevic, Stankovic e Stojcev, 2014). Poucas implementações, contudo, permitem cobertura completa de todos os casos possíveis de falhas e muito esforço tem sido aplicado para resolver este problema incluindo métodos de síntese lógica de FSM que já incluam os componentes para auto verificação (Garvie e Husbands, 2019).

2.14 FPGA (Field Programmable Gate Array)

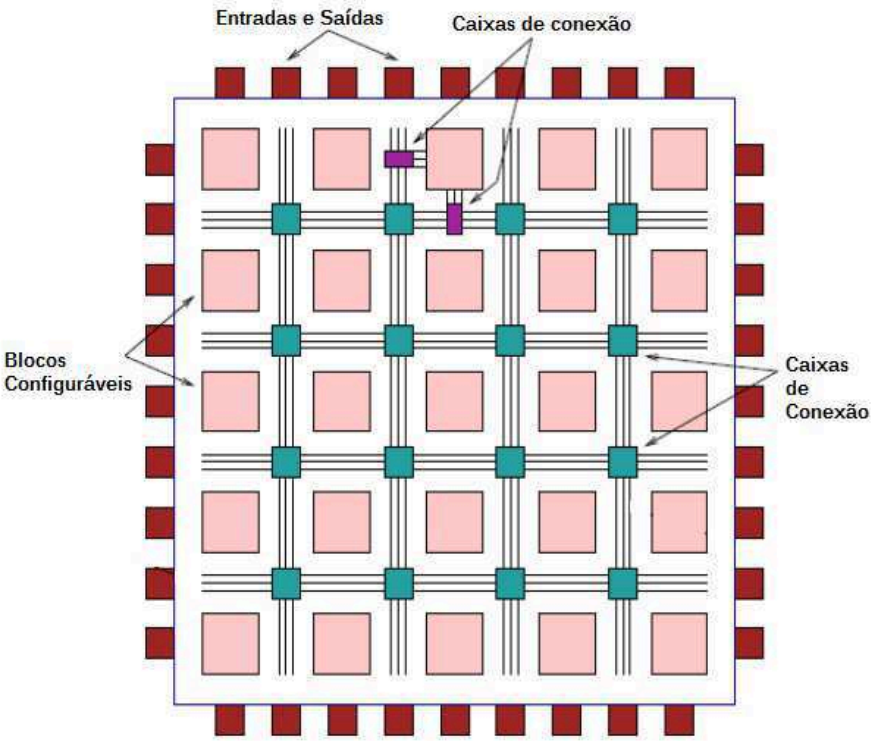
Os FPGAs são componentes eletrônicos contendo uma matriz de circuitos lógicos que podem ser interligados de diversas formas, de acordo com dados guardados numa memória. Por isso são chamados de matriz de portas digitais programável em campo. O termo "em campo" significa que podem ser programadas depois de saírem da fábrica de circuitos integrados, onde foram produzidos. As falhas por SEE são especialmente críticas em FPGAs devido à sua natureza volátil: além das memórias e registradores presentes nos circuitos implementados num FPGA, a própria memória que determina como os circuitos são interconectados pode sofrer uma alteração quando atingida por uma partícula de alta energia (Yuan, Zhao-wei e Dan, 2008).

Os FPGAs são circuitos integrados contendo uma grande quantidade de elementos de circuitos digitais, como *flip-flops* e portas lógicas, permeadas por linhas de interconexão. Uma memória de configuração é utilizada para indicar como estes elementos são ligados nas linhas de interconexão para formar circuitos complexos (Piltan *et al.*, 2016). A Figura 20 mostra como é uma estrutura básica de um FPGA que emprega caixas de conexão para interligar os blocos de lógica configuráveis.

Em um certo tipo de FPGA a memória de configuração é construída com fusíveis, dificultando sua alteração por causa de um SEU. Este modelo de FPGA tem sido preferencial nas aplicações aeroespaciais. Contudo, a grande maioria dos FPGAs de alta capacidade, atualmente disponíveis, empregam uma memória RAM (*Random Access Memory*) e esta pode ser alterada por um SEU.

Num exemplo de circuito em um FPGA, se a memória de configuração indica que uma operação AND deve ser feita com dois sinais, após um SEU o circuito pode mudar para uma operação XOR, causando uma falha operacional. Esta falha só pode ser detectada se esta memória de configuração for examinada frequentemente comparando com alguma referência. Esta comparação é o método mais frequentemente empregado para resolver problemas causados por SEU em FPGAs. Uma vez descoberta a falha o sistema é reinicializado forçando a memória de configuração do FPGA a ser renovada e corrigida. Esta solução, contudo, não impede alguma operação crítica ser realizada antes que a falha seja detectada, nem evita a perda de tempo e de possíveis informações quando o sistema precisa ser reinicializado.

Figura 20 - Estrutura básica de um FPGA



Fonte: Modificado de (Piltan *et al.*, 2016)

Uma falha de configuração em um FPGA é denominada de SEFI (*Single Event Functional Interrupt*) (Gaillard, 2011)(Wang e Agrawal, Vishwani D, 2008)(Trivedi e Mehta, 2016). Esta sensibilidade dos FPGAs levou muitos pesquisadores a desenvolver técnicas baseadas em TMR para mitigar estes problemas, sem contudo obter uma solução completa (Wirth, Kastensmidt e Ribeiro, 2008) (Kastensmidt *et al.*, 2005).

Muitas pesquisas têm sido feitas sobre a sensibilidade de FPGAs a falhas por SEE, incluindo TID, porque estes dispositivos são utilizados frequentemente em sistemas aplicados em satélites e sondas espaciais, além de pesquisas científicas e militares. Foi demonstrado experimentalmente que o acumulo de radiação (TID), quando expondo aos raios X, resultou no aumento do consumo de corrente de um FPGA proporcionalmente à dose irradiada (Silveira *et al.*, 2015). A partir deste resultado pode-se extrapolar que, num ambiente continuamente exposto à radiação, um FPGA vai se degradar de forma cumulativa até que se torne inoperante.

O estudo feito por Silveira mostrou que após a detecção de uma falha causada pela radiação, e mantendo o FPGA temporariamente sem energia, ocorre o processo de *annealing*, mesmo em temperatura ambiente, onde parte dos danos causados por TID são eliminados. Isso indica que, com o tempo, certos defeitos causados por TID podem ser corrigidos, quando a

estrutura cristalina do Silício tem condições de recuperar sua forma original. Este processo é o mesmo que ocorre quando os fabricantes de circuitos integrados aquecem o substrato de Silício para permitir que a estrutura cristalina possa se reordenar após sessões de implantação de dopantes e em outras etapas da fabricação.

Este processo de *annealing* sugere que, uma aplicação em satélite com FPGAs, poderiam ser empregdos circuitos redundantes em que apenas um esteja energizado, de cada vez. Quando uma falha ocorrer o outro circuito é ligado e aquele que falhou é desligado, permitindo que o processo de *annealing* ofereça um certo nível de cura e talvez possa ser utilizado novamente, aumentando a vida útil do satélite.

3 CRITÉRIOS PARA PROJETO DE EQUIPAMENTO AERONÁUTICO

A motivação para este estudo veio da necessidade de empregar tecnologia adequada no desenvolvimento de equipamentos aeronáuticos, cujas exigências de qualidade, confiabilidade e segurança são muito elevadas. Para certas categorias de equipamentos, como aqueles responsáveis pela segurança, a ocorrência de falhas que possam resultar em perdas de vida deve ter uma probabilidade menor do que uma em 10^9 horas de voo (dependente do tipo de aeronave; em aeronave com peso menor que 2700 kg e um motor a hélice o mesmo equipamento precisaria garantir apenas 10^6 horas de voo sem falhas). Isso coloca a possibilidade de ocorrência de um SEU ou SET, em certos circuitos, como sendo algo relevante que precisa ser evitado.

Além da exigência na confiabilidade e segurança existe o fator econômico. Não é economicamente viável desenvolver circuitos integrados utilizando técnicas RHBD, especificamente para uso aeronáutico, quando a aplicação é em sistemas comerciais (não militar). Sistemas militares costumam ter orçamentos que suportam os custos de desenvolvimento e certificação de circuitos integrados com desenhos específicos, mas sistemas comerciais precisam empregar componentes já existentes no mercado (chamados de COTS: *Commercial Off-The-Shelf* ou um item comercial de prateleira) ou fabricar circuitos integrados empregando bibliotecas de desenho padronizadas (comerciais). Isso torna quase impossível encontrar componentes que já integrem em seu projeto as técnicas RHBD descritas anteriormente. Por isso o interesse em pesquisar arquiteturas de circuitos que possam lidar com ocorrências de SEU e SET, mantendo os resultados confiáveis e que possam ser produzidos empregando tecnologias comerciais.

Durante o período deste estudo o autor participou do desenvolvimento de um equipamento aeronáutico para controle de Flaps *(Flap Control Unit*, FCU), seguido alguns meses depois por um controlador ambiental (*Air Conditioning Control Unit*, ACCU). O FCU controla a posição de certas superfícies de voo que auxiliam nos pousos, decolagens e em manobras avançadas durante o voo. O ACCU controla a temperatura na cabine tanto em solo como em grandes altitudes, quando é necessário pressurizar a cabine, em voo noturno ou diurno, resfriando-a com o uso de um sistema de ar-condicionado ou aquecendo com o uso do sistema chamado *Bleed*. O *Bleed* é um fornecimento de ar pressurizado e superaquecido, proveniente

do motor, que não pode ser liberado descontroladamente na cabine, sob o risco de causar ferimentos por queimadura na tripulação.

Equipamentos assim, que possam causar ferimentos ou afetar o voo da aeronave, possuem critérios muito restritos para seu desenvolvimento. As autoridades certificadoras, como a ANAC, no Brasil, só permitem que tais equipamentos sejam utilizados se comprovarem ter a confiabilidade e segurança necessárias. Para tanto, existem normas para os métodos, processos e acompanhamento do desenvolvimento, sendo as mais conhecidas a norma DO-254 para projetos eletrônicos e a DO-178 para projetos de programação de computadores. Esta última deve ser cumprida sempre que algum tipo de computador, microprocessador ou componente considerado complexo (como certos FPGAs) é empregado nos sistemas vitais do equipamento sendo desenvolvido. A certificação pela DO-178 é muito rigorosa. Apenas o custo das ferramentas de verificação (já certificadas) e do pessoal necessário para elaborar todos os documentos de um equipamento, como por exemplo um rádio digital, ultrapassa a cifra de 1 milhão de dólares e demanda não menos que 12 meses de trabalho (Thomas, 2009).

Considerando os limites orçamentários para o desenvolvimento dos equipamentos citados, e visando obter a certificação sem precisar da DO-178, a escolha sobre o caminho a ser seguido recai em empregar componentes classificados como simples. Evita-se o uso de microcontroladores tradicionais e utilizam-se circuitos lógicos convencionais com a funcionalidade dos requerimentos do sistema. Deste modo o circuito pode ser classificado, conforme a DO-254, como um SEH (*Simple Electronics Hardware*) para poder ser desenvolvido e aprovado a um custo menor.

A norma DO-254 trata das fases do processo de desenvolvimento de um equipamento, estabelecendo que, ao seguir os critérios e processos corretos, o equipamento obtido sempre é desenvolvido de um modo que garante a melhor confiabilidade e segurança possíveis. O processo de desenvolvimento e posterior certificação empregam como guia, além de outras normas, as recomendações expostas no documento CAST-30 (*Certification Authorities Software Team*, CAST) (CAST, 2007) ou mesmo no documento da EASA AMC 20-152 (EASA, 2018). Nestes documentos são esclarecidos os critérios que uma entidade certificadora deve considerar para admitir, com segurança, que um determinado dispositivo tenha atendido os requisitos para ser classificado como SEH.

Esta classificação é aceita, mesmo para dispositivos configuráveis como os FPGAs, desde que atenda um requisito específico: um dispositivo é considerado simples quando um

conjunto coerente de testes e análises determinísticas puderem demonstrar seu funcionamento correto em todas as condições operacionais previsíveis, sem qualquer comportamento anormal (CAST, 2007). Isso implica que é preciso verificar a correta operação quando todas as possíveis combinações, permutações e condições concorrentes tanto nas entradas de sinal como nos circuitos internos sejam exercitadas. Um FPGA pode conter diversos circuitos lógicos independentes e cada um deles deve ser verificado e atender os critérios de SEH individualmente para que todo o componente seja aceito no projeto.

A Figura 21 mostra a aparência dos equipamentos FCU e ACCU, além de exibir a aeronave B-250 onde foram instalados e testados. Esta aeronave é empregada para instrução e acrobacias em altitudes até 30.000 pés (9 Km). A pressurização é necessária a partir de 10.000 pés (3 Km).

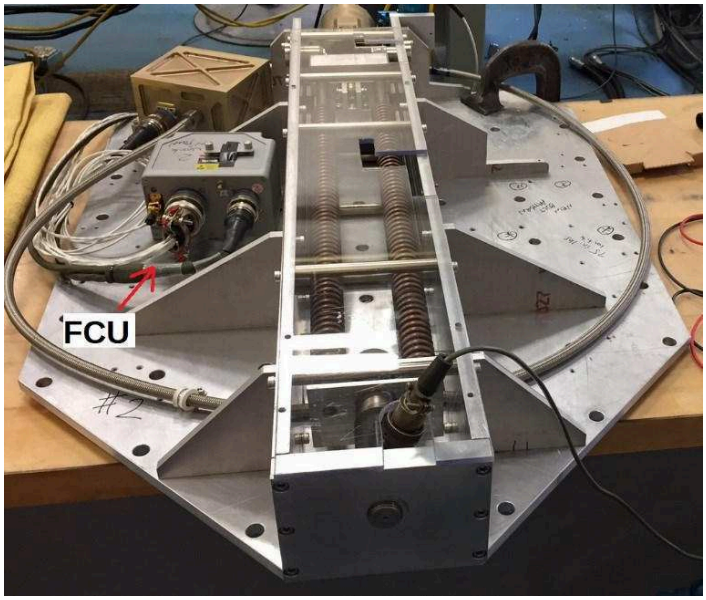
Figura 21 – FCU, ACCU e a aeronave B-250

Fonte: Autor

No caso do FCU, para esta aeronave especificamente, o funcionamento é aparentemente bastante simples: uma alavanca próxima ao piloto possui 3 posições, uma para cada ângulo dos Flaps que podem ser utilizados durante o voo: 0°, que recolhe os Flaps e é usada durante o voo; 15°, que permite aumentar levemente a superfície das asas dando maior sustentação e é utilizada na decolagem; e 40°, que permite voo em baixa velocidade, adequada para o pouso. O FCU opera atuando um motor elétrico que move os Flaps. Este motor consome cerca de 45 Ampères e o sistema precisa funcionar num ambiente com temperaturas variando de -54°C até 70°C. Na Figura 22 é mostrada a bancada de testes do FCU onde um sistema mecânico reproduz as cargas aerodinâmicas previstas que podem ocorrer nos Flaps em todas as

condições possíveis de voo. A segunda caixa mostrada na figura contém o motor elétrico que aciona o atuador dos Flaps através de um eixo flexível.

Figura 22 - Bancada de testes do FCU



Fonte: Autor

Diferente de outras aeronaves que usam sensores de posição discretos para saber onde os Flaps estão, nesta aplicação alguns sensores fornecem leituras contínuas com valores angulares entre 0 e 41°, com resolução de 0,1°. Portanto, o circuito de controle precisa comparar a posição real dos Flaps com a posição esperada, selecionada na alavanca de comando e decidir quando deve mover os Flaps. Uma dificuldade a mais, que motivou o uso de um FPGA (para ser programável em campo), está em que os ângulos dos Flaps, para as 3 posições da alavanca de comando, possam ser mudados depois do equipamento ser fabricado. Isso acontece porque após os ensaios de voo os pilotos de teste podem querer mudar estes ângulos para valores mais adequados, segundo a performance aerodinâmica da aeronave. Por exemplo, mudar o ângulo de decolagem de 15° para 18°. Isso implica que tais valores não poderiam ser fixos no circuito de controle e sim programáveis.

O circuito de controle da posição dos Flaps é considerado um SEH, compreendendo unicamente um comparador de magnitude (lógica combinacional) de valores binários entregues pelos sensores de posição e um registrador que contém o valor do ângulo que os Flaps devem ser posicionados. Se a posição corrente dos Flaps for maior do que o ângulo informado no registrador o motor será ligado para retrair os Flaps até que a comparação mostre igualdade.

Nos requerimentos do projeto foi incluída uma função que evita mover os Flaps depois que estiverem na posição escolhida até que o piloto selecione uma posição diferente. Assim, quando o comparador resulta em igualdade o motor é desligado até que uma nova posição seja escolhida. Esta é uma lógica muito simples e o controle completo pode ser verificado integralmente.

O projeto então resultou num circuito híbrido: um comparador de posição para saber se os Flaps deveriam avançar ou recuar; e uma máquina de estados tipo PSM para gravar e ler dados de uma memória não volátil, escrevendo estes dados no registrador do comparador conforme a seleção feita na alavanca de controle. A PSM é também utilizada para testar os circuitos eletrônicos continuamente e transmitir pacotes de dados para o computador da aeronave, usando o protocolo ARINC-429 (meio de comunicação de dados através de dois fios e que é um padrão utilizado em aeronaves). Estes dados são usados para poder mostrar a situação dos Flaps em algum painel acessível à tripulação.

Como a PSM faz o tratamento da memória não volátil onde as posições programadas para os Flaps são guardadas, torna-se importante que ela seja confiável e segura, não podendo gravar ou ler dados incorretos sem detectar esta situação. As exigências de confiabilidade levaram a este estudo, para poder determinar de que forma uma falha por SEU ou SET poderia afetar o sistema, e qual arquitetura digital poderia ser empregada para reduzir ou eliminar os riscos durante seu funcionamento.

4 DESENVOLVIMENTO DE SOLUÇÕES

Apesar das possibilidades de aumento da imunidade à SEE empregando técnicas de RHBD, os projetos de equipamentos aeronáuticos convencionais (não militares ou para uso espacial/científico) acabam ficando limitado ao uso de circuitos integrados já existentes no mercado. Isso impõe condições de projeto em que o circuito precisa conviver com a ocorrência de SEU ou SET sem sofrer falhas de sistemas. Em outras palavras, os circuitos devem poder sofrer um SEU ou SET e conseguir resolver uma falha interna sem causar uma falha de sistema.

Neste capítulo são discutidas propostas de solução para ocorrências de falhas por SEU e SET.

4.1 Arquitetura PSM tolerante a falhas por SEU e SET

Uma máquina de estados finita programável, PSM, funciona como um computador convencional empregando uma memória para armazenar os estados possíveis. Este tipo de circuito traz grande flexibilidade ao permitir adaptar o projeto de uma máquina de estados sem precisar alterar o circuito fisicamente (Luba, Gorski e Wronski, 1992) (Garcia-Vargas *et al.*, 2007). A maior confiabilidade e segurança obtidas com uma PSM resistente a falhas por SEU ou SET pode contribuir para o emprego destes circuitos em aplicações críticas.

A arquitetura proposta foi apresentada no Simpósio Latino Americano de Testes realizado em 2018 (LATS2018) (Einsfeldt, A. e Giacomini, 2018) e permite resolver uma falha, em um bit da PSM, no intervalo de tempo de um estado desta máquina. O trabalho apresentado encontra-se no Apêndice A. A arquitetura descrita neste capítulo possui algumas diferenças, decorrentes do estudo desenvolvido depois da publicação e que permitiu aperfeiçoar alguns aspectos da tecnologia.

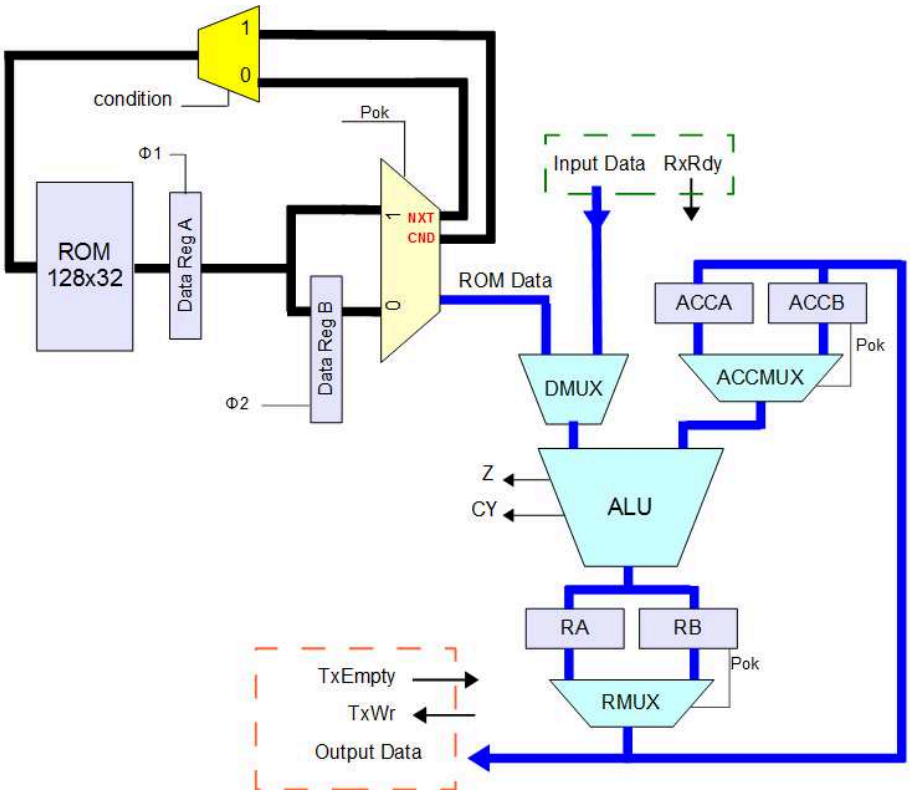
A PSM aqui apresentada emprega uma memória não volátil (*Read Only Memory*, ROM) com paridade embutida para armazenar as possíveis saídas para cada estado e o endereço da memória para o próximo estado.

O diagrama da PSM é apresentado na Figura 23. Sua estrutura é similar a um microprocessador convencional, porém sem o contador de programa. Estão presentes uma

Unidade Lógica Aritmética (*Arithmetic and Logic Unit*, ALU), registradores para operações como o Acumulador (ACC) e Resultado (R), além de seletores de barramentos e outros elementos comuns à CPUs de microcontroladores. A arquitetura efetivamente implementada nos equipamentos aeronáuticos é mais complexa, além de lidar com certas interfaces como uma memória EEPROM para guardar dados de voo e algumas entradas e saídas analógicas do sistema.

O aspecto mais importante da PSM proposta é limitar o número de próximos estados possíveis em apenas dois, agrupando estes valores bem determinados junto com os dados lidos da ROM. Isso simplifica a implementação das medidas contra os efeitos causados por SEU e SET e resulta numa máquina de estados mais confiável. Quando o valor do próximo estado (endereço na memória ROM) é fixado desde o momento da compilação do programa fica mais fácil introduzir nos dados da memória algum método de verificação embutido nos próprios dados.

Figura 23 - Arquitetura da PSM



Fonte: Autor

Este tipo de tecnologia e arquitetura não foram encontradas em trabalhos recentes de outros pesquisadores nem em circuitos aplicados comercialmente. Um conceito similar foi empregado nos primeiros computadores desenvolvidos a partir de 1950, como no IBM-650

(https://en.wikipedia.org/wiki/IBM_650) ou no RPC-4000, em 1960, da empresa norte-americana Royal McBee. No IBM-650 e no RPC-4000 cada posição da memória de programa continha a instrução a ser executada e o endereço que deveria ser usado para encontrar a instrução seguinte.

Conforme novas tecnologias apareceram a arquitetura de computadores mudou e o endereço das instruções passou a ser armazenado num registrador chamado Contador de Programa (*Program Counter*, PC), também chamado de Apontador de Instruções *(Instruction Pointer*, IP), porque era uma forma de economizar espaço da memória e permitia uso de circuitos para armazenar endereços de sub-rotinas. O uso do PC nas arquiteturas de computadores se tornou um padrão e virtualmente todos os estudos envolvendo máquinas de estado ou CPUs possuem, atualmente, registradores para armazenar o estado (endereço) sendo executado. Contudo, tal padrão de arquitetura perdeu sua maior confiabilidade inerente a um programa que contém seu próprio endereçamento. Como informação complementar, o IBM-650 foi a série de computadores mais popular da IBM sendo o primeiro modelo a ser produzido em massa no mundo em 1953, com preço inicial de US$47.000. Pesava mais de 2 toneladas e possuía uma memória magnética com apenas 2000 posições e valores de 10 dígitos, tanto para dados como para programa, construída num tambor giratório que permitia tempo de acesso para cada dado de apenas 4 milissegundos (muito rápido para a época).

Diferentes desenhos de PSM podem ser implementados empregando as mesmas técnicas de tolerância às falhas descritas neste trabalho.

Assume-se que o conteúdo da ROM não é modificável por um SEU ou SET, mesmo considerando que algumas memórias reprogramáveis, como EEPROM e FLASH, comercialmente disponíveis, podem ser sensíveis a elevadas doses de radiação (Layton, Longden e Patnaude, 2005). Uma memória ROM de grande confiabilidade pode ser obtida empregando o tipo anti-fusível (*antifuse*) ou OTP (*One Time Programmable*) (McCollum *et al.*, 2002). Estas memórias, quando adquiridas comercialmente, são fornecidas apagadas pelo fabricante e podem ser alteradas uma única vez, pelo usuário. Depois de terem sido gravadas não podem mais ser alteradas. O processo de gravação destrói o caminho físico elétrico que determina o estado 1 ou 0 de um bit, de modo que não é possível alterar seu estado posteriormente. Compradas como circuitos integrados individuais, este tipo de ROM é chamada de PROM (*Programmable Read Only Memory*), mas neste trabalho, por simplificação, será chamada apenas como ROM.

Numa PSM convencional, o próximo estado é a combinação do estado atual presente nos dados armazenados na ROM com os sinais de entrada (Garcia-Vargas *et al.*, 2007). Isso implica o uso de uma quantidade variável de *flip-flops* para armazenar as possíveis combinações em cada estado. Na arquitetura proposta, o endereço para o próximo estado fica permanentemente embutido nos dados da ROM, em dois segmentos destes dados, limitando o valor (endereço) do próximo estado em apenas duas possibilidades: uma quando uma condição interna é verdadeira e a outra quando é falsa. Como estes valores são garantidos através da verificação de integridade, a confiabilidade para o circuito escolher corretamente o próximo estado fica limitada ao sinal de condição. Este sinal é obtido através de um circuito combinacional e por isso fica sujeito apenas aos efeitos momentâneos de um SET.

A composição do conteúdo dos dados na ROM compreende os dois endereços possíveis para o próximo estado, alguns bits para dados e outros para sinais que vão controlar as ações da máquina de estados. Esta distribuição do conteúdo da ROM está mostrada na Figura 24. Para limitar a largura da ROM em 32 bits foi escolhido um barramento de endereços de apenas 7 bits (bits 0 a 6), suficiente para 128 posições de memória, simplificando o desenvolvimento de um sistema para o teste prático. A paridade embutida nos endereços para o próximo estado está no oitavo bit (bits 7 e 15 na Figura 24) de cada segmento dos dados utilizados para este fim. Nas implementações práticas feitas, para os equipamentos aeronáuticos citados, a ROM tem 64 bits de largura e os endereços tem 9 bits, permitindo programas com até 512 instruções.

Figura 24 - Barramento da memória ROM

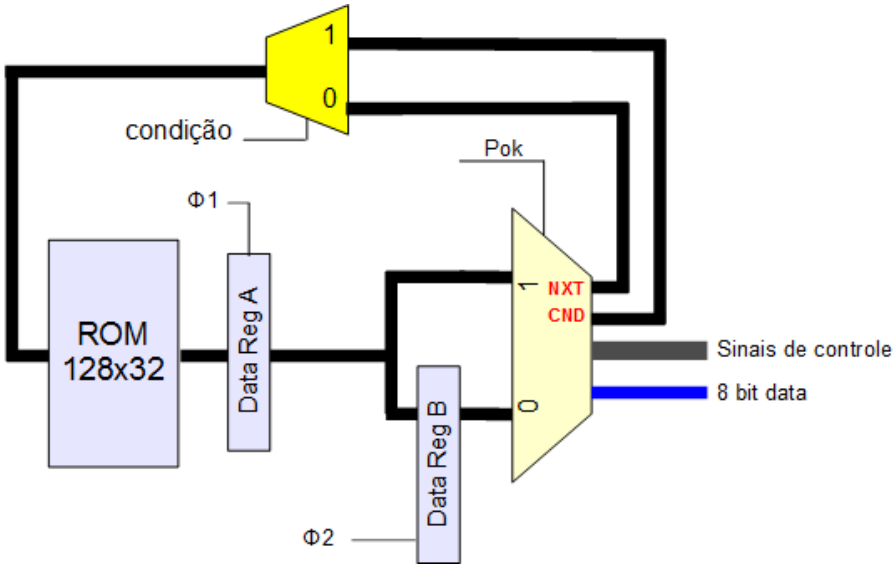
ROM 32 bit de largura de dados			
31 – 24	**23 – 16**	**15 – 8**	**7 – 0**
linhas de controle	8 bit dados	próximo endereço se a condição for verdadeira	próximo endereço se a condição for falsa

Fonte: Autor

Dois registradores armazenam o conteúdo completo dos dados da ROM, em redundância. O esquema deste setor do circuito está apresentado na Figura 25.

O sinal Pok, nesta figura, indica se a paridade do Data Reg A está correta, após uma porta XOR calcular a paridade de todos os bits, incluindo o oitavo bit que é a paridade embutida nos dados da ROM. Este cálculo de paridade verifica se o que foi escrito no registrador permanece correto, comparando os bits de informação com o bit de paridade que vem dos dados da ROM. O mesmo é feito com o registrador Data Reg B, mas este sinal de teste de paridade é utilizado apenas para o caso de ambos os registradores estarem com paridade incorreta.

Figura 25 - Barramento de endereço



Fonte: Autor

O uso de registros para armazenar os dados da ROM é necessário porque a memória, tipicamente, não possui um registrador próprio. Muitas das memórias ROM (ou PROM) convencionais possuem apenas circuitos combinacionais de decodificação de endereço. Isso significa que alterando um dos bits de endereço causa uma alteração de qual parte da memória será lida e transferida para o barramento de dados de saída, tão rápido quanto o tempo de propagação destes circuitos. Se alguns bits dos dados da memória fossem conectados ao barramento de endereço, esta realimentação faria a decodificação interna ficar mudando a posição lida tão rápido quanto possível, sem controle.

Quando a PSM está em um estado qualquer o registrador Data Reg A mantém as informações válidas para aquele estado. Como mostrado na Figura 25, dois grupos de bits na saída do seletor de dados fornecerão os endereços possíveis para o próximo estado (exibidos na figura como NXT e CND, acrônimos adotados para "*Next*", ou "Próximo", e "Condicional").

Ambos seguem o caminho de retorno para endereçar a ROM, mas apenas um deles é selecionado pelo sinal de condição. Seja qual for o novo endereço, ele vai apontar na ROM uma nova posição, contendo um novo conjunto de dados que será mais tarde registrado no Data Reg A, novamente, e assim por diante.

Os registradores Data Reg A e Data Reg B são acionados por sinais de *clock* que ocorrem em momentos diferentes, Φ1 e Φ2. O *clock* Φ2 ocorre depois do Φ1. Então, o Data Reg B somente será carregado com a informação presente no registrador Data Reg A depois que este tiver carregado os dados da ROM e somente se a paridade do registrador A (sinal Pok na figura) estiver correta.

Se ocorrer um SEU num bit de Data Reg A antes dos dados serem guardados em B a operação do estado atual da PSM é cancelada e um novo Φ1 é aguardado para corrigir os dados em A através de uma nova escrita nele. Isso é possível porque nenhuma operação foi executada ainda (os demais *clocks* ainda não ocorreram). Esta funcionalidade pode ser obtida usando o próprio sinal Pok, como qualificador para gerar o *clock* Φ2 e os demais *clocks* do circuito da PSM. Se o sinal Pok indicar um erro de paridade depois de Φ2 o seletor de dados comuta para o Data Reg B, pois este deve conter dados corretos (presumindo apenas um bit de toda a PSM, no Data Reg A, ter sido modificado por um SEU).

O sinal de condição, empregado na escolha do endereço do próximo estado, é resultante de operações internas, como quando o resultado da ALU é zero ou se existem dados presentes na entrada do circuito. A decisão de usar um ou outro endereço para o próximo estado funciona como um Pulo Condicional, muito conhecido em programação de microcomputadores.

A ocorrência de um SET neste sinal de condição poderia causar uma perturbação na PSM caso tenha duração maior que o tempo de acesso da memória (t_{ROM}) somado com o tempo de propagação do seletor multiplex (t_{MUX}) e o tempo de acerto (*setup time*) do registrador Data Reg A (t_{SA}). Portanto, uma falha pode ocorrer se a duração do SET for grande o suficiente, como expresso na Inequação 7:

$$t_{SET} > t_{ROM} + t_{MUX} + t_{SA} \tag{7}$$

Contudo, um SET tem duração (t_{SET}) típica menor que 1 nanosegundo (Mohanram, 2005) (Balasubramanian *et al.*, 2008). Por isso, a memória ROM escolhida deve ter tempo de acesso muito maior do que isso para evitar uma perturbação por este efeito. Isso vale também

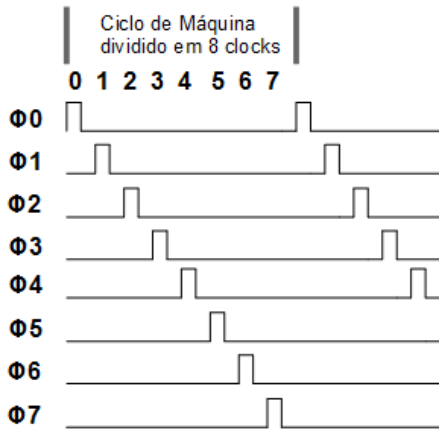
para um SET que ocorra em alguma das linhas de endereço de memória, logo depois do seletor de condição.

Em alguns casos este tipo de procedimento sequencial de correção de falhas pode causar um atraso no processo sendo tratado pela PSM, quando um novo Φ1 precisa ser aguardado. Atrasos assim podem não impactar na performance da PSM num sistema pois ocorrem raramente, por natureza, e com curta duração (frações de microssegundo, dependendo da velocidade do *clock* do sistema). A detecção e correção de falha acontece apenas no caso de SEU e SET, ou seja, apenas um bit alterado diretamente num registrador ou um pulso transiente que ocorra em um só sinal do circuito causando uma alteração de um bit. Eventos múltiplos, que podem não ser detectados através do teste de paridade, por exemplo, poderiam ser resolvidos empregando um tipo de verificação mais sofisticado.

Na arquitetura proposta e no circuito implementado para testes, as linhas de *clock* totalizam oito sinais com duração pouco menor que 1/8 do tempo de ciclo de máquina (Figura 26). O ciclo de máquina é assim denominado como a duração de todos os processos contidos em um estado da PSM. Cada sinal está atrasado em 1/8 do ciclo de máquina, exibindo, portanto, uma defasagem entre cada uma.

Estas linhas de *clock* defasadas, presentes nos registradores de dados da ROM e outros registradores do circuito da PSM, têm a limitação de exigir que o intervalo entre os *clocks* seja maior do que a duração esperada de um SET. Deste modo os efeitos de um SET ficam limitados a um só domínio de tempo.

Figura 26 – Representação da distribuição dos *clocks* defasados



Fonte: Autor

Existem 8 sinais de *clock*, e 8 habilitadores (*clock enable*). A finalidade de usar um habilitador para cada *clock* em vez de apenas um *clock* contínuo e comum a todos, é aumentar a imunidade a falhas por SET. Se um SET ocorrer num sinal de *clock enable* de um *flip-flop*, no momento em que o *clock* comum estiver ativo, ele vai registrar um dado na hora errada. A geração de *clocks* defasados exige um circuito como um *shift-register* (registrador de deslocamento) e *buffers* (amplificadores) para a distribuição destas linhas de *clock* entre todos os registradores da PSM. Isso, contudo, faz parte das técnicas para mitigar os efeitos de eventos únicos em arquiteturas de circuitos digitais.

O emprego de clocks isolados por portas lógicas (*clock gating*) é uma técnica que está sendo utilizada para reduzir a potência consumida por microprocessadores (Yamada *et al.*, 2006) e a distribuição sugerida aqui, com 8 *clocks* defasados independentes, tem o mesmo efeito.

Uma simulação feita com SPICE de uma carga submetida a um *clock* contínuo e outro descontínuo, ativo por 1 ciclo a cada 8, demonstrou não apenas uma redução de 8 vezes na energia como também um espalhamento maior no espectro de frequência. Este espalhamento resulta na redução de RFI (*Radio Frequency Interference*) que é muito conveniente em componentes para aplicação aeroespacial. A emissão reduzida de campos eletromagnéticos é um fator importante nestas aplicações.

Mesmo contando com os *clocks* defasados a possibilidade de uma falha causada por estas linhas deve ser tratada. Por isso, para evitar que um SET possa causar falhas ao atingir um determinado sinal de *clock*, foram incluídos os 8 habilitadores (*clock enable*), especificamente para deste modo exigir, em cada *flip-flop*, dois sinais independentes para seu acionamento.

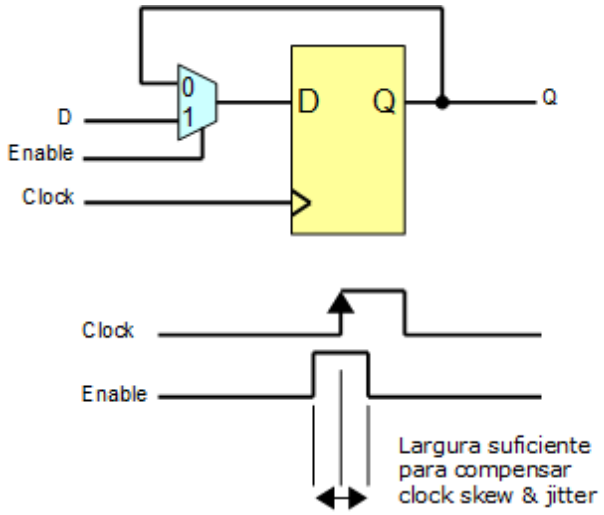
Simbolicamente o *clock enable* autoriza, ou não, que um novo dado seja armazenado apenas quando ocorre um evento de *clock*, empregando para isso um selecionador (*multiplex*), como representado na Figura 27. Portanto, se o *clock enable* não estiver ativo quando ocorre o evento de *clock*, o *flip-flop* apenas registrará o dado já armazenado nele, sendo insensível ao valor do dado na entrada. Tal método é comum no desenho de circuitos integrados digitais e os dois sinais podem ser distribuídos fisicamente distantes um do outro para reduzir a interferência mútua.

O *clock enable* deve ser produzido por um circuito da PSM levando em consideração os requisitos de *setup time* (tempo da chegada do dado antes do evento de *clock*) do *flip-flop*,

jitter (variação de período) e *skew* (atraso de propagação) normalmente existentes na malha de distribuição dos sinais de *clock* para os registradores (*flip-flops*).

Deste modo um SET em um dos sinais (*clock* ou *clock enable*) não se propaga para todos *flip-flops* existentes na malha de *clock* porque, naquele instante, o outro sinal não deve estar também ativo. Se o outro sinal estiver ativo, então a ocorrência do SET não causa falhas, pois acontece no momento em que o registrador deveria memorizar, normalmente.

Figura 27 – Uso de *clock enable*

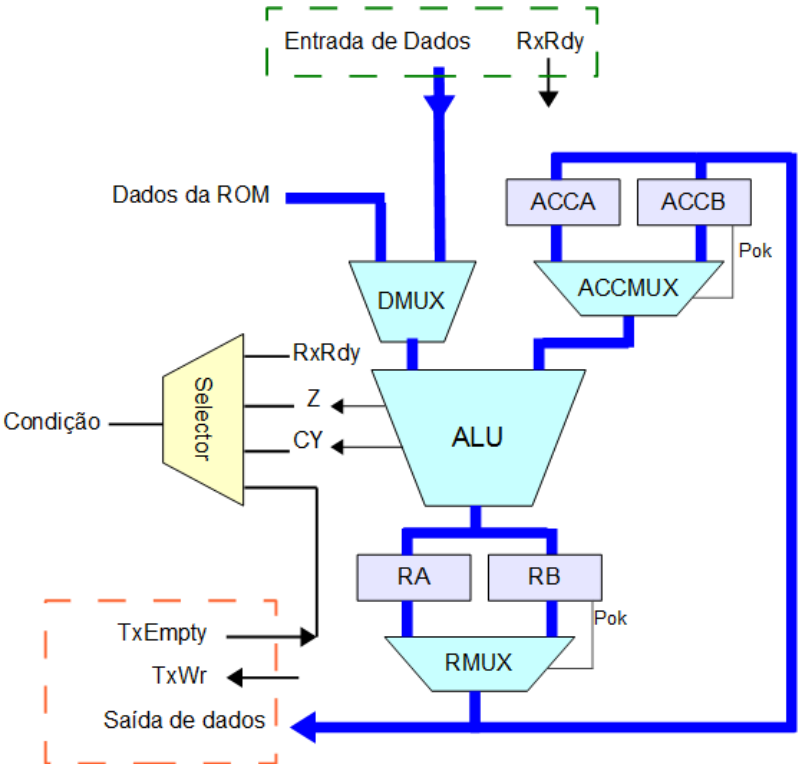


Fonte: Autor

A seguir é analisado o circuito de dados desta PSM, mostrado na Figura 28, estes passam obrigatoriamente pela ALU e depois todos os resultados são armazenados em RA e RB. A ALU possui duas entradas, sendo uma de dados selecionados por DMUX e a outra do Acumulador (ACC). O acumulador é exibido como dois registradores, ACCA e ACCB, porque são redundâncias. Do ponto de vista lógico existe apenas um Acumulador e é assim denominado ao logo do texto.

O Acumulador é alimentado pelo resultado das operações da ALU através de RA e RB (que são redundantes, armazenando o resultado da ALU).

Figura 28 - Barramentos de dados



Fonte: Autor

Na arquitetura desenvolvida para este trabalho poucas operações foram implementadas na ALU, suficientes para testar adequadamente a arquitetura proposta. Estas operações estão listadas na Tabela 1.

Tabela 2 - Operações da ALU

Operação	Descrição
1	Guarda em $R_{(AB)}$ o dado em DMUX
2	Guarda em $R_{(AB)}$ o resultado da função DMUX xor ACC
3	Guarda em $R_{(AB)}$ o resultado da função DMUX and ACC
4	Guarda em $R_{(AB)}$ o resultado da função aritmética DMUX-ACC
5	Em qualquer das operações anteriores executa uma rotação para a esquerda no resultado antes de memorizar em $R_{(AB)}$

Fonte: Autor

São operações lógicas e matemáticas que permitem testar a imunidade a falhas causadas em qualquer bit dos registradores, ou alterações de processo resultantes de transientes em qualquer sinal sensível. Esta finalidade será descrita no item 4.1.2.

Para as condições que levam à seleção do endereço do próximo estado foram escolhidas 4 possibilidades listadas na Tabela 2.

Tabela 3 - Condições para seleção de endereço

Condição	**Descrição**
1	Presença de dados externos
2	Resultado zero na ALU
3	Bit de carregamento (Cy) em 1
4	Transmissão de dados na saída foi completada (*TxEmpty*)

Fonte: Autor

Após a ALU processar uma operação a paridade do resultado é calculada e armazenada, com o resultado, em dois registradores, RA e RB. São empregados sinais de *clock* defasados, tal como foi feito com os registradores dos dados da ROM (ver Figura 25). Se, depois de memorizar o resultado em RA e RB, um deles for afetado por um SEU a paridade verificada vai indicar a falha e o seletor RMUX selecionará o registrador com os dados corretos. Esta duplicação de registradores consome alguma área de Silício e energia, porém muito menos do que resultaria com a triplicação como em um TMR (Kumar, Chellappa e Clark, 2015).

Outra medida de proteção foi adotada adicionando-se um comparador para os dados armazenados em RA e RB, que devem ter sempre o mesmo conteúdo. Isso é feito porque um SET, ocorrendo na lógica que seleciona a operação em andamento na ALU ou em algum componente da ALU, poderia causar um resultado com paridade válida na saída da ALU, mas com conteúdo diverso do outro registrador que armazenaria o resultado em momento diferente. Neste caso, a diferença detectada faz o circuito da PSM ignorar todos os demais *clocks*, fazendo com que o sequenciamento do estado corrente seja repetido completamente. Como no caso da

falha por SEU no Data Reg A, citado como exemplo anteriormente, um evento assim resolve a falha sem danos à missão, além de um atraso equivalente à duração de um estado da PSM . Este esquema com comparador não é necessário nos registradores da ROM nem no Acumulador porque os dados neles armazenados contém o bit de paridade já embutido tanto na ROM como em RA e RB.

4.1.1 Caso especial: Inicialização

Considerando uma aplicação prática desta PSM é preciso tratar seu funcionamento durante a energização do circuito, quando a PSM vai começar a funcionar num estado qualquer. Este tratamento é considerado um caso especial pois ocorre apenas quando o circuito é energizado.

O primeiro estado a ser executado pode ser sempre um estado conhecido, como o do endereço 0 da memória, porém dependendo da programação da PSM é possível fazer a máquina identificar sua situação presente neste momento e adotar um próximo estado consistente. Fazendo a máquina de estados executar sempre um determinado primeiro estado torna a programação mais simples.

Na PSM proposta e desenvolvida para testar as premissas de resolução de falhas por SEU e SET, o primeiro estado é o de número 0 (zero). A garantia de que seja sempre este estado foi obtida empregando portas AND em todos os bits do barramento de endereço que vai para a memória ROM. Uma das entradas de cada porta AND é ligada no bit correspondente vindo do seletor de condição. A outra entrada é ligada em comum a um sinal de inicialização que, além disso, também controla o circuito gerador de *clock*. Para evitar a sensibilidade a um SET justamente num sinal tão importante, este controle é separado em dois sinais independentes, unidos junto a cada porta AND através de uma porta OR. Estes sinais permanecem normalmente em nível lógico 1. Seria preciso que ambos sinais estivessem em nível lógico 0, simultaneamente, para as portas AND ficarem com 0 em suas saídas.

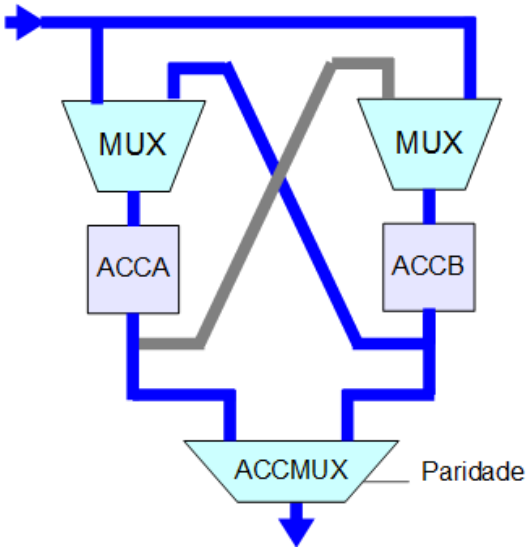
Este circuito de inicialização não foi incluído nas Figuras 23 e 25 para manter os diagramas mais simples.

4.1.2 Recuperação de Dados no Acumulador

O resultado da ALU pode eventualmente ser escrito no Acumulador. Quando isso é feito seus dois registradores redundantes, ACCA e ACCB, recebem o dado e o memorizam empregando dois sinais de *clock* defasados para assim obter também uma redundância temporal. Se um dos registradores sofrer um SEU o circuito de paridade identifica o problema e o seletor de dados na saída do Acumulador (ACCMUX) vai manter os dados corretos disponíveis para uso pela PSM.

Contudo, o dado armazenado no Acumulador pode permanecer nele por um tempo indeterminado e, por isso, seu conteúdo precisa ser garantido indefinidamente. Se ocorrer um SEU num dos registradores ele deve ser reescrito, assim que possível, usando o dado íntegro existente no outro registrador. Esta atividade é feita empregando seletores de dados nas entradas de cada registrador de modo a permitir obter o dado correto sempre que necessário. A Figura 29 mostra este esquema. A lógica de controle dos seletores depende da detecção de falha de paridade em cada registrador e de um outro *clock*. Este é utilizado para realizar a operação de reescrita o mais cedo possível, reduzindo a chance de um novo SEU ocorrer no registrador com o dado correto antes dele ser usado para reescrever no registrador que havia falhado.

Figura 29 - Esquema de recuperação de dados



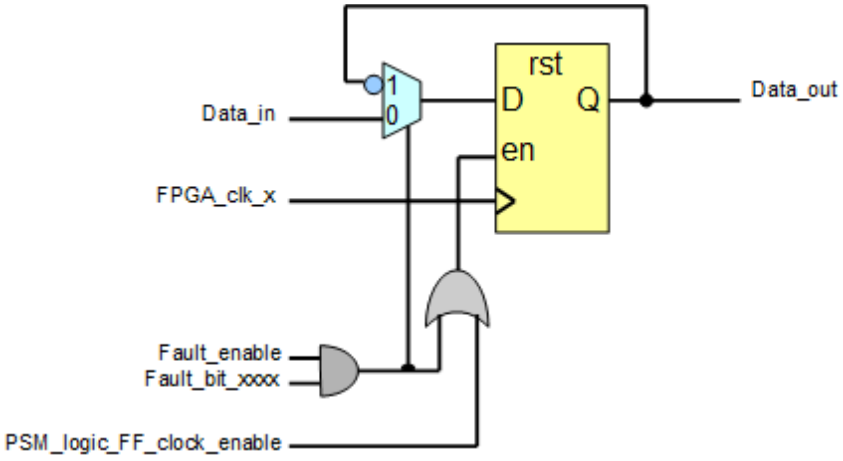
Fonte: Autor

4.1.3 Verificação da Tolerância da Arquitetura Proposta

O método de verificação considera que a arquitetura proposta deve se recuperar de um SET ou SEU que tenha ocorrido no circuito. Isso torna o valor do LET_{th} irrelevante para verificação de capacidade de recuperação, pois não se discute a sensibilidade a um evento único, mas a capacidade do circuito para manter a integridade dos dados e sua funcionalidade após a ocorrência de um SET ou SEU. Além disso, todas as portas lógicas e *flip-flops* devem ter desenho convencional e não empregam nenhuma técnica de reforço contra efeitos da radiação (RHBD) para evitar mudança de bits ou propagação de transientes.

Foi desenvolvido um circuito equivalente em um FPGA, de forma que, além da PSM, estivessem também presentes circuitos para injeção de falhas simulando os efeitos de SET e SEU. Para isso, os *flip-flops* usados na PSM foram cercados de uma lógica como mostrado na Figura 30 e que é apropriada para os testes em FPGA.

Figura 30 - Esquema de um *flip-flop* da plataforma de teste



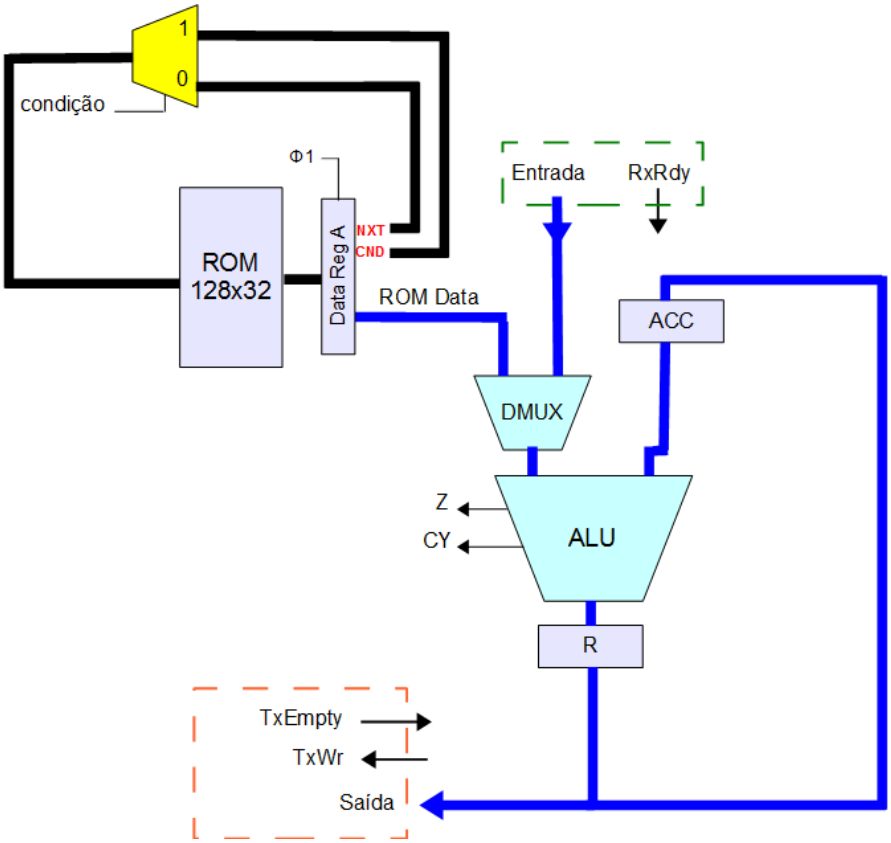
Fonte: Autor

Esta lógica permite simular um *bit flip* (inversão de estado do bit memorizado no *flip-flop*) quando estão ativos os sinais que habilitam uma falha (*Fault_enable* e *Fault_bit_xxxx*). O sinal *FPGA_clk_x* é um sinal do sistema que ocorre nas fases previstas para os *clocks* defasados da PSM (*x* é de 0 a 7). O sinal *PSM_logic_FF_clock_enable* é o sinal *clock enable*, descrito no texto da PSM (ver Figura 27).

O circuito completo do injetor de falhas funciona empregando um registrador de deslocamento com 120 bits e o sinal de habilitação (*Fault_Enable*) que é ligado com a duração de um dos *clocks* defasados e em um dos 8 momentos existentes em um ciclo de máquina. Deste modo é possível injetar uma falha em qualquer *flip-flop* ou linha de *clock*, exatamente no momento mais crítico, obtendo resultados confiáveis sobre a vulnerabilidade de um circuito.

Os testes para verificação da vulnerabilidade foram feitos empregando duas implementações da arquitetura desta PSM: uma versão contendo os circuitos de tolerância a falhas e outra versão sem eles. A versão sem os circuitos de tolerância a falhas contém apenas um registrador de cada tipo e seu diagrama está apresentado na figura 31.

Figura 31 - PSM sem os circuitos de tolerância a falhas



Fonte: Autor

Ambas versões da PSM processam um programa que executa um algoritmo gerador de números pseudoaleatórios, cuja semente é o valor de um byte entregue como dado de entrada nesta máquina de estados finitos. Este algoritmo é executado 12 vezes, de forma consecutiva, na memória da PSM e assim permite ocupar 127 das 128 posições possíveis da memória ROM.

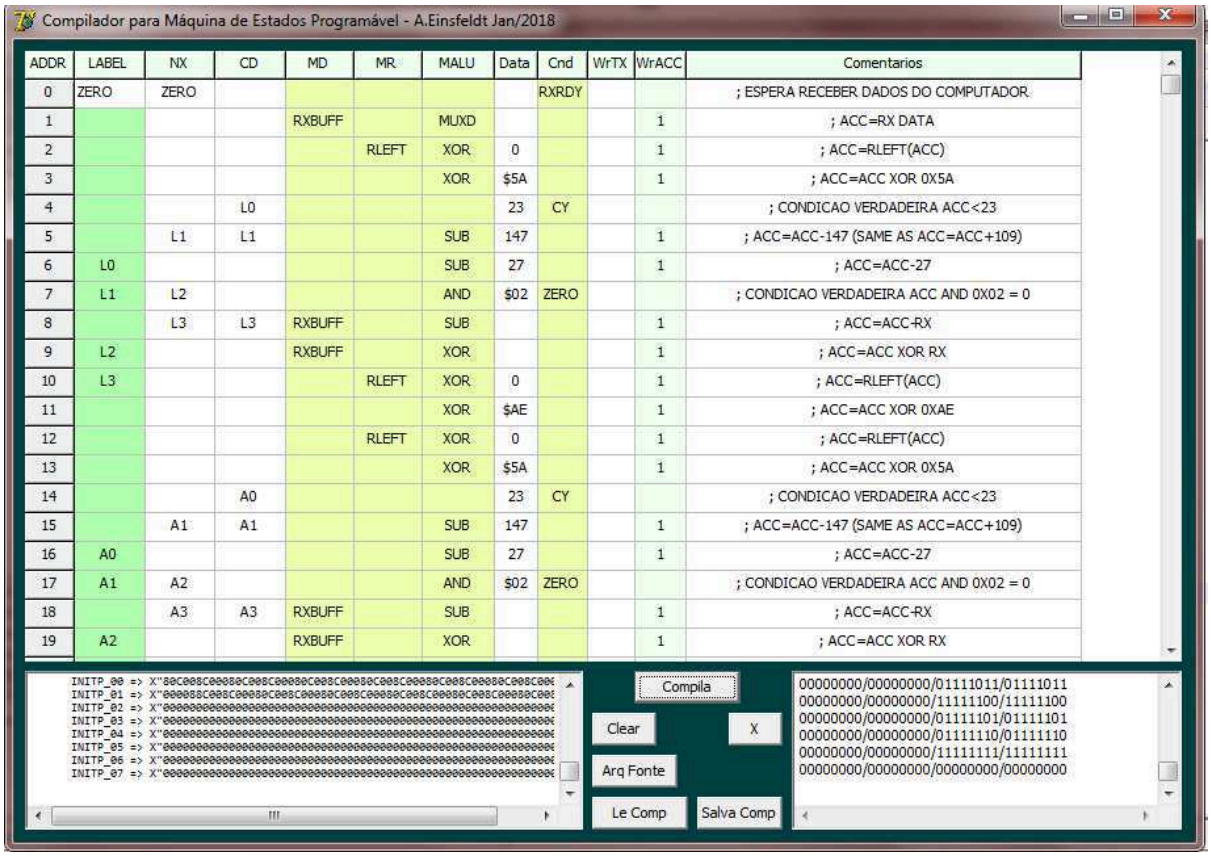
Por exemplo, iniciando com o dado 0x02 cada um dos estágios seguintes vai resultar nestes valores, em sequência: 0x3D, 0xB9, 0x87, 0x2F, 0x61, 0xA8, 0x73, 0xE0, 0xA2, 0xBB, 0x9F e finalmente 0x0F. Este último valor é transmitido para fora da PSM, para ser comparado com o que seria esperado e, deste modo, confirmar se não houve alguma corrupção de informação ou do processo em todos os 12 estágios. Um único bit diferente em algum estágio contaminaria os próximos causando diferença no resultado. Como base de referência para comparação, quando nenhuma falha é injetada o resultado deve ser sempre o mesmo em ambas implementações.

A entrada e saída de dados, bem como o registrador de deslocamento para injeção de falhas, são controlados por uma interface de comunicação com um computador. Adicionalmente, foi construído um contador de *clocks*. Ele serve para indicar em qual *clock*, no conjunto completo da execução dos 12 estágios geradores de números, o sinal de falha deve ser injetado. No computador é executado um programa que envia o número inicial para a PSM, o número do *clock* de quando a falha deve ser injetada e também um pacote com todos os bits do registrador de deslocamento para indicar em qual (ou quais – no caso de teste com múltiplos bits de falha simultânea) sinal interno a falha deve ocorrer. Este programa de supervisão do teste no computador pode verificar a ocorrência de falhas não recuperáveis tanto de forma aleatória como exercitando cada sinal e cada momento de *clock* possível.

4.1.4 Compilador de linguagem

Um compilador foi especialmente desenvolvido pelo autor para facilitar a criação de programas para esta PSM. Um arquivo de texto é lido e convertido nos dados a serem gravados na memória ROM. Esta facilidade permitiu o uso de rótulos (*labels*) para as linhas com as instruções eliminando o trabalho tedioso de contar as linhas e atualizar os valores dos endereços dos estados, além de calcular automaticamente a paridade dos endereços. Na Figura 32 é mostrada a aparência do compilador e as primeiras linhas do programa de este da PSM.

Figura 32 - Compilador para a PSM



ADDR	LABEL	NX	CD	MD	MR	MALU	Data	Cnd	WrTX	WrACC	Comentarios
0	ZERO	ZERO						RXRDY			; ESPERA RECEBER DADOS DO COMPUTADOR
1				RXBUFF		MUXD				1	; ACC=RX DATA
2					RLEFT	XOR	0			1	; ACC=RLEFT(ACC)
3						XOR	$5A			1	; ACC=ACC XOR 0X5A
4			L0				23	CY			; CONDICAO VERDADEIRA ACC<23
5		L1	L1			SUB	147			1	; ACC=ACC-147 (SAME AS ACC=ACC+109)
6	L0					SUB	27			1	; ACC=ACC-27
7	L1	L2				AND	$02	ZERO			; CONDICAO VERDADEIRA ACC AND 0X02 = 0
8		L3	L3	RXBUFF		SUB				1	; ACC=ACC-RX
9	L2			RXBUFF		XOR				1	; ACC=ACC XOR RX
10	L3				RLEFT	XOR	0			1	; ACC=RLEFT(ACC)
11						XOR	$AE			1	; ACC=ACC XOR 0XAE
12					RLEFT	XOR	0			1	; ACC=RLEFT(ACC)
13						XOR	$5A			1	; ACC=ACC XOR 0X5A
14			A0				23	CY			; CONDICAO VERDADEIRA ACC<23
15		A1	A1			SUB	147			1	; ACC=ACC-147 (SAME AS ACC=ACC+109)
16	A0					SUB	27			1	; ACC=ACC-27
17	A1	A2				AND	$02	ZERO			; CONDICAO VERDADEIRA ACC AND 0X02 = 0
18		A3	A3	RXBUFF		SUB				1	; ACC=ACC-RX
19	A2			RXBUFF		XOR				1	; ACC=ACC XOR RX

Fonte: Autor

O resultado do compilador é um arquivo de texto escrito em representação Hexadecimal (caracteres 0 até F) que logo a seguir é carregado em uma outra ferramenta que integra o projeto completo e programa o FPGA na placa eletrônica.

A título de exemplo é apresentado a seguir um segmento do arquivo de texto que representa o algoritmo (depois repetido 12 vezes) a ser executado pela PSM durante o teste funcional:

```
:zero   nx=zero        cnd=rxrdy                    ; Espera receber dados do computador
                       malu=muxd md=rxbuff wracc    ; ACC=RX data
                       malu=xor data=0 mr=rleft wracc ; ACC=RLEFT(ACC)
                       malu=xor data=$5A wracc      ; ACC=ACC xor 0x5A
        cd=L0          cnd=cy data=23               ; Condição verdadeira se ACC<23
        nx=L1 cd=L1    malu=sub data=147 wracc      ; ACC=ACC-147
:L0                    malu=sub data=27 wracc       ; ACC=ACC-27
:L1     nx=L2          malu=and data=$02 cnd=zero   ; Condição verdadeira se ACC e 0x02 = 0
        nx=L3 cd=L3    malu=sub md=rxbuff wracc     ; ACC=ACC-RX
:L2                    malu=xor md=rxbuff wracc     ; ACC=ACC xor RX
:L3                    malu=xor data=0 mr=rleft wracc ; ACC=RLEFT(ACC)
                       malu=xor data=$AE wracc      ; ACC=ACC xor 0xAE
```

Os detalhes da linguagem de programação não serão incluídos aqui pois se trata, apenas, de uma ferramenta desenvolvida para permitir executar os testes e demonstração funcional. Porém, cabe apontar a versatilidade de uma PSM como esta que possui um sequenciamento interno para a movimentação de dados dentro de um estado. Por exemplo, na segunda linha (estado da PSM) o programa comanda a ALU para transferir os dados de MUXD para sua saída (malu significa: Modo ALU). Na mesma linha (no mesmo estado) o MUXD (indicado como md) deve usar o dado recebido pelo computador (rxbuff). Ainda no mesmo estado é comandado que os dados presentes nos registradores $R_{(AB)}$ na saída da ALU devem ser escritos no Acumulador (na instrução wracc). Portanto, nesta linha é executada uma operação que lê o dado recebido do computador e o transfere para o Acumulador através da ALU.

4.1.5 Método de teste

A verificação da vulnerabilidade da arquitetura desta PSM deve ser feita injetando falhas em todos os pontos em que um SEU ou SET poderiam resultar em erro não recuperável. Estes pontos são o conteúdo de cada flip-flop utilizado nesta PSM, os sinais de *clock* e os sinais combinacionais que possam interferir no funcionamento da PSM. Alguns sinais não foram incluídos no teste porque não têm relevância pois a injeção de falhas neles não poderia resultar em erro. Os sinais incluídos e excluídos foram:

- Registradores (Data Reg A, Data Reg B, RA, RB, ACCA, ACCB): Estes são os únicos elementos que contém *flip-flops* na PSM. Qualquer bit deles, inclusive o bit reservado para a paridade, deve poder receber uma falha injetada.

- Seletor de funções da ALU: Este sinal combinacional. quando alterado por um SET no momento em que ocorre um sinal de *clock* nos registradores RA ou RB, pode armazenar um valor completamente diferente do esperado porque a função sendo calculada pela ALU seria incorreta. Este é um sinal importante e uma injeção de falha que mude a função da ALU representa o mesmo efeito do que um SET em qualquer um, das dezenas de nodos que compõe o circuito da ALU.

- Sinal de Condição: Este é considerado um sinal crítico pois uma alteração nele num momento próximo a um sinal de *clock* pode comprometer qual dado será lido da ROM. A injeção de falha neste sinal, contudo, simula a duração de um SET centenas de vezes maior do

que ocorre na realidade e, portanto, impõe uma condição de falha definida e que deve também ser tratada pelo circuito da PSM.

- Sinais de *clock*: Cada um dos sinais de *clock* defasados e utilizados para sequenciamento do processo da PSM possui duas vias de transmissão (o *clock* e o *clock_enable*), separadas fisicamente. Esta técnica resolve o problema de propagação de um SET quando existe uma linha de *clock* única, como explicado anteriormente. Um SET só pode ocorrer em uma das linhas e não existindo a outra ativa ele não se propaga para os *flip-flops*. Ainda assim, para demonstrar que esta lógica é imune, foram incluídas duas injeções de falha nestes sinais.

- Resultados dos cálculos de paridade: Estes sinais não receberam injeção de falhas. Os cálculos de paridade só podem ter duas saídas possíveis, paridade correta ou incorreta. Injetar uma falha para indicar paridade incorreta tem exatamente o mesmo efeito que existir, de fato, uma paridade incorreta em um registrador. Isso apenas acionaria o mecanismo de resolução do problema, mesmo que não exista um problema realmente. Por outro lado, dentro da premissa que esta arquitetura resolve falhas ocorridas em um único bit, não existiria caso em que haveria uma falha de paridade (algum bit de um registrador alterado por SEU ou SET) e uma outra falha causar, simultaneamente, um erro de cálculo de paridade impedindo da falha ser detectada. O teste injetando falhas nos registradores já cobre a ocorrência de uma falha na saída de um teste de paridade.

- Sinais que determinam a condição (linhas de endereço para o próximo estado): Estes sinais vão determinar a condição, como um resultado zero na ALU ou o bit de carregamento (Cy) ligado. Eles não recebem injeção de falha porque a consequência seria a mesma que injetar um SET no sinal de condição, como já foi considerado anteriormente.

- Sinais do comparador de RA e RB: Não possuem injeção de falhas porque sua consequência seria indicar uma diferença entre RA e RB mesmo quando ela não existe. Neste caso este seria o mesmo efeito de quando existe uma diferença causada por uma mudança na função da ALU no momento em que um dos registradores vai guardar o resultado da ALU. Este caso está coberto pela injeção de falha no seletor de funções da ALU.

- Sinais do barramento de dados de entrada: Possuem injeção de falhas para cobrir a ocorrência de SET nos circuitos fora da PSM. Contudo, estas falhas devem ser detectadas pelo circuito de comparação de RA e RB pois um SET nos sinais de dados implicaria num resultado da ALU diferente do esperado.

Cada um dos sinais que aceitam injeção de falha deve receber a falha 8 vezes a cada ciclo de máquina para coincidir com os 8 *clocks* gerados defasados. Isso garante que as falhas sejam injetadas nos momentos mais críticos. Falhas tipo SET ocorrendo em momentos fora da ocorrência de um *clock* não causam erros nos circuitos.

Injetar as falhas, apenas nos momentos em que um evento possa causar um problema, implica que o teste é mais rigoroso do que se o circuito fosse exposto a um fluxo de partículas de alta energia.

Na arquitetura proposta os 12 estágios de programa na ROM consomem 90 ciclos de máquina, cada um. Isso resulta em 720 ocorrências dos oito *clocks* defasados. Para testar os efeitos de SET ou SEU ocorrendo em cada sinal sensível (120 ao todo) e em cada instante possível (720) são necessários 86.400 testes. Na versão implementada sem os circuitos de tolerância a falhas, o número de *flip-flops* e sinais críticos diminui e a quantidade de sinais sensíveis totaliza 53. Um teste completo de todos os sinais nos momentos possíveis para esta implementação exige 38.160 testes (720x53).

4.1.6 Resultados dos Testes

Na tabela 2 são apresentados os resultados dos testes realizados. Foram incluídos testes com ocorrência randômica apenas para demonstrar que o método é validado estatisticamente. Nota-se que, para a arquitetura sem tolerância a falhas, os erros ocorrem em 28% das vezes. Isso foi porque algumas das falhas injetadas ocorreram em momentos ou em sinais que não causavam erros no processo. Esta alta porcentagem demonstra como o processo é sensível. Por outro lado, empregando os circuitos de tolerância a falhas o resultado é zero erros, ou seja, mesmo que um SET ou SEU ocorram em qualquer sinal sensível e crítico a PSM continuará a funcionar sem exibir erros.

Tabela 4- Resultados dos testes em FPGA

Arquitetura	Método	Falhas injetadas	Falhas não resolvidas	Taxa de Falhas
Tolerante a falhas	Varredura de todas as falhas possíveis	86400	0	0
Não tolerante a falhas	Varredura de todas as falhas possíveis	38160	10865	28%
Não tolerante a falhas	Aleatório	82926	23920	29%

Fonte: Autor

Testes com dupla injeção de falha foram feitos embora a arquitetura proposta tenha sido elaborada para tolerar apenas falhas únicas. Os resultados, como esperado, foram menos satisfatórios exibindo 1814 erros de processo em 48104 falhas duplas injetadas aleatoriamente. Portanto, para cada 26 falhas duplas causadas por partículas energéticas o circuito apresentou um resultado incorreto (3,8%).

Já a comparação do uso de recursos indicou que a arquitetura proposta implica em 89% a mais de componentes do FPGA do que a versão sem os circuitos de tolerância a falhas. Apesar de parecer um número alto ainda, está muito menor que a utilização típica acima de 200% quando se emprega TMR e abaixo da duplicação simples e comparação (que seria resultaria em mais de 100% de recursos). A quantidade de recursos do FPGA pode ser equiparada em magnitude com a área de Silício caso um circuito integrado utilize esta tecnologia e, portanto, observa-se a economia quando empregando a tecnologia proposta.

A distribuição dos *clocks* defasados também contribui na redução do consumo de energia, podendo representar uma economia acima de 50% conforme a quantidade de registradores conectados em cada sinal de *clock*.

4.1.7 Considerações sobre Melhorias

Uma análise adicional foi proposta para considerar uma situação em que o circuito eletrônico tenha sido implementado utilizando tecnologias avançadas, de alta velocidade, para verificar se existem casos em que alguma falha não detectável possa ocorrer. Os casos possíveis envolvem alguns sinais nos dados da ROM que são responsáveis por selecionar funções da lógica da PSM.

Por exemplo, considerando um caso em que um bit dos dados no Registrador A for modificado de 0 para 1, resultando na mudança de operação na ALU de lógica XOR para AND. Este bit será propagado do Registrador A até a ALU e vai perdurar pelo tempo de propagação do circuito detector de paridade. Se a duração do bit com valor incorreto for tão longa quanto o tempo de operação da ALU, pode ser que um dado calculado como AND em vez de XOR esteja presente na entrada de RA ou RB. Se estiver coincidente com o *clock* de um destes registradores, um dado falso será armazenado com a paridade correta.

Para evitar este tipo de falha, mesmo que de reduzida probabilidade, foi incluído na arquitetura um comparador entre os dados de RA e RB. Se o RA registrar um dado incorreto por causa da situação citada acima, o RB vai armazenar o dado correto porque seu *clock* ocorre num outro instante. A diferença entre os dois registradores vai causar uma parada da PSM, resultando em executar novamente o estado atual. Este modelo de resolução do problema serve para todos os sinais de seleção de função que, neste desenho de PSM, levam sempre para um mesmo resultado ser armazenado em dois momentos diferentes. Um projeto de PSM diferente da proposta precisa levar em conta a possibilidade destas ocorrências.

4.2 Registrador Auto Recuperável Empregando Amplificador *Tri-State*

Após o desenvolvimento da arquitetura de PSM tolerante a falhas por SET ou SEU, o próximo passo foi aperfeiçoar o circuito do registrador de dados chamado de Acumulador. Este trabalho foi publicado no SEMINATEC 2018 sob o título "*Self-Recoverable Data Register Under Presence of SEE*" (Einsfeldt, Augusto e Giacomini, 2018). O propósito do estudo publicado foi simplificar o circuito do Acumulador da PSM. Na arquitetura original ele exige um seletor de dados e um circuito lógico para manter os dados corretos, sobrescrevendo algum dos registradores caso apresente algum erro de paridade. O desenvolvimento de melhores soluções para este circuito traz redução de área de silício e energia, tornando o projeto mais eficiente e com menor probabilidade de falhas.

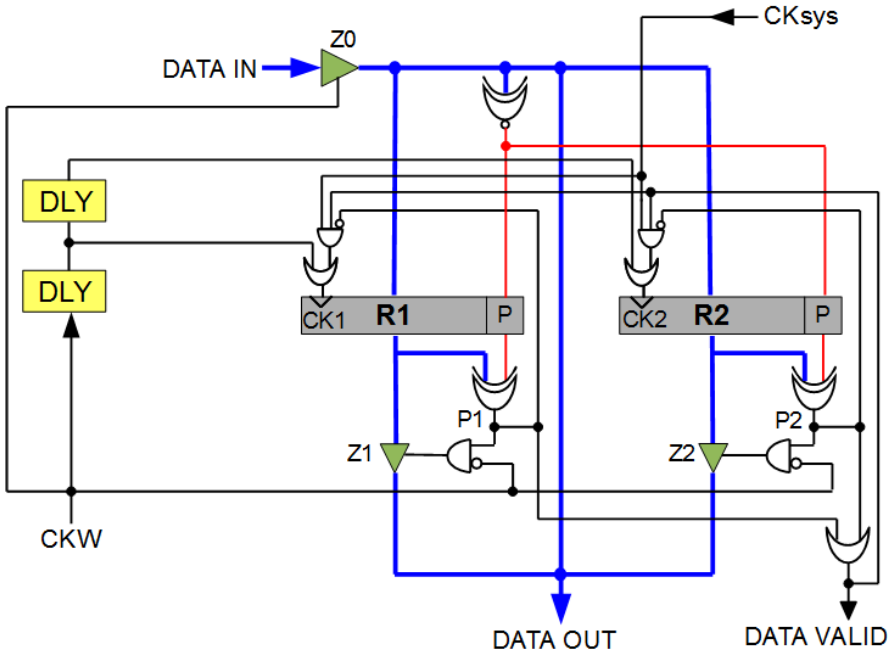
Além da aplicação como Acumulador na PSM, este circuito de registrador auto recuperável foi considerado, de forma independente, para uso no projeto do Controlador de Flaps, para melhorar a confiabilidade dos registradores que guardam a posição que os Flaps

devem estar. Desta forma estes registradores são mantidos íntegros independentemente de uma supervisão da PSM.

O circuito proposto permite manter dados armazenados, com integridade, após a ocorrência de um SET ou SEU, empregando, para isso, uma verificação desta integridade e uma cópia do registrador de dados. Uma importante diferença em relação ao circuito na PSM apresentado anteriormente é o emprego de amplificadores (*buffer*) *tri-state* (cuja saída pode adotar os estados lógicos 0 ou 1, além de também o estado Z, de alta-impedância). Com componentes *tri-state* obtém-se uma grande redução dos recursos utilizados (área de Silício) e simplificação no processo de sobrescrever quando ocorre alguma falha em um registrador, reduzindo o tamanho do circuito de suporte e controle.

Foi também incluído um circuito para o cálculo de paridade. Na PSM a paridade já é calculada antes do armazenamento em RA e RB, tornando desnecessário calcular novamente no Acumulador. Este registrador auto recuperável, contudo, foi pensado para uso em qualquer aplicação e por isso incorpora o cálculo de paridade. O circuito completo é apresentado em grande detalhe na Figura 33. Quando fazendo parte da PSM, este circuito de cálculo de paridade poderia ser retirado para economizar área de Silício.

Figura 33 - Circuito para detecção e correção de falhas por SET ou SEU



Fonte: Autor

Os amplificadores *tri-state* (triângulos verdes) são utilizados normalmente como meio para construção econômica de multiplexadores (Alioto, Cataldo, Di e Palumbo, 2002), além de permitir a conexão entre dois barramentos.

Durante o funcionamento normal aparece um dado na entrada (DATA IN) para ser armazenado nos registradores. Isso vai ocorrer quando o sinal de *clock* CKW é ativado em 1, acionando o *tri-state* Z0 e bloqueando os *tri-states* Z1 e Z2 (a Figura 34 mostra o diagrama de tempo dos sinais relevantes). O *clock* CKW é o sinal que causa o processo de gravação dos dados nos registradores.

Esta operação tem dois efeitos: primeiro, faz o barramento de dados ser disponibilizado imediatamente por Z0 no barramento de saída DATA OUT, enquanto desliga a saída dos registradores que, a esta altura, possuem dados antigos. Isso é útil para circuitos de alta velocidade, em que os dados a serem gravados precisam também estar disponíveis para uso o quanto antes possível.

O segundo efeito é que já se inicia o cálculo da paridade dos dados após Z0 (linha vermelha no desenho). Assim, depois de um pequeno atraso (primeiro bloco DLY) o sinal de *clock* CKW estará acionando a escrita no registrador R1, já com o valor de paridade estável.

Em mais alguns instantes, após o segundo bloco DLY, o sinal CKW, um pouco mais atrasado, chega para a escrita no registrador R2. Este duplo atraso tem a finalidade de escapar de um SET que possa atingir algum dos bits de dados no exato momento em que o sinal CKW chega nos registradores. Uma diferença de tempo permite que um, ou outro, registrador armazene os dados corretos. Se o SET ocorrer na linha de paridade (vermelha) apenas um dos registradores armazenará a paridade incorreta. Esta técnica, empregando atrasos (DLY) para lidar com um SET, é frequentemente empregada em circuitos DTMR (Hasanbegovic e Aunet, 2016)(Kastensmidt *et al.*, 2005) e em *flip-flops* projetados para suportar SET (Kumar, Chellappa e Clark, 2015).

A duração do atraso dos blocos DLY deve ser igual ou maior que o tempo de propagação da porta OR, na entrada de *clock* dos registradores, mais o tempo exigido de chegada dos dados nos registradores antes do evento de *clock* (*setup time*). Esta duração não pode ser inferior à duração estimada de um SET.

Enquanto o sinal CKW estiver ativo, além dos registradores serem escritos, o barramento de saída DATA OUT será mantido com o mesmo valor de DATA IN. Quando

CKW é desligado os amplificadores Z1 e Z2 vão assegurar que os mesmos dados continuem sendo apresentados em DATA OUT, pois foram armazenados em R1 e R2. Tanto Z1 como Z2 permanecem ligados enquanto as paridades de R1 e R2 estiverem corretas. Se um dos registradores for atingido por uma partícula de alta energia, mudando seu conteúdo, a verificação de paridade vai desligar o amplificador *tri-state* correspondente.

Enquanto pelo menos um dos registradores estiver com paridade correta o sinal DATA VALID permanecerá ligado informando o restante do sistema que os dados podem ser utilizados. Na eventualidade de ambos os registradores serem atingidos, tanto Z1 como Z2 ficarão desligados e o sinal DATA VALID será também desligado.

Importante notar que na ocorrência de um SEU em um dos registradores os dados na saída dele estarão incorretos durante o tempo de propagação do circuito verificador de paridade e desligamento do *tri-state* correspondente. Durante este tempo pode ocorrer uma situação de contenção de barramento (naquele bit que foi alterado pelo SEU) que tipicamente resulta em um pico de consumo de corrente (porque, naquele bit, cada componente *tri-state* estará com a saída em polaridade oposta ao outro). Portanto, é recomendável utilizar transistores de saída para o *tri-state* que sejam fracos, ou seja, possuam pouca capacidade de corrente. Assim, um caso destes resultaria em uma corrente apenas moderada com duração de poucos picosegundos. No item 4.3 é discutida uma outra forma de implementar este tipo de circuito sem o risco de transientes de corrente.

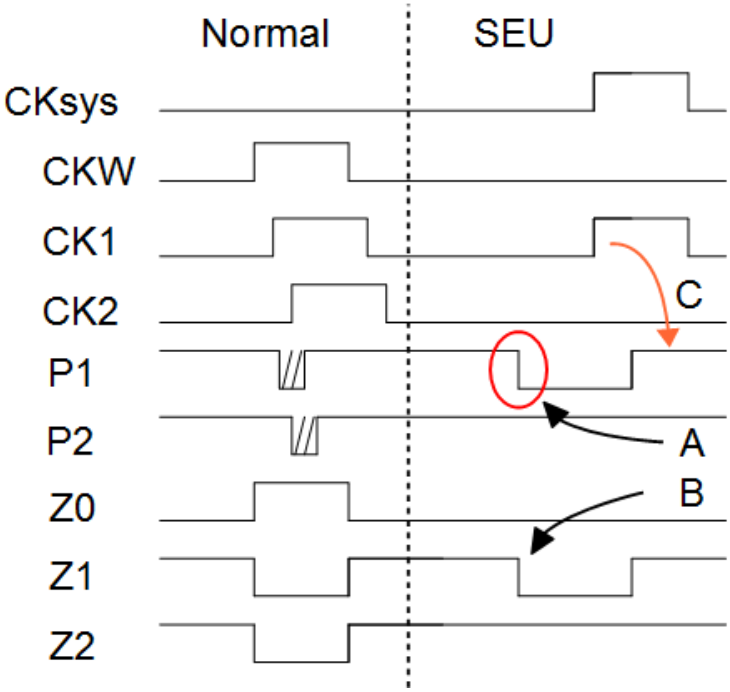
Quando um dos registradores entra em falha por SEU um circuito de recuperação é habilitado pela detecção de diferença na paridade esperada. Neste circuito o barramento de saída DATA OUT está também presente nas entradas dos registradores R1 e R2. Na ocorrência de uma falha por SEU, o *tri-state* correspondente é desligado, mantendo corretos os dados presentes nas entradas de R1 e R2. A falha de paridade habilitará a escrita no registrador com dados incorretos empregando um sinal de *clock* auxiliar (CKsys).

Este *clock* auxiliar provém de algum sinal externo a este circuito e, se for conveniente ao projeto, pode estar ativo mais frequentemente do que o CKW para garantir pronta recuperação dos dados, antes mesmo que um novo SEU possa degradar as informações do outro registrador. Importante que CKsys não seja assíncrono em relação à CKW porque não deve haver situação em que ele possa coincidir com CKW dentro do intervalo de tempo correspondente ao atraso equivalente a um DLY. Uma coincidência nestes dois sinais de *clock*,

caso um dos registradores apresente uma falha detectada pela paridade, poderia resultar em armazenar dados antes deles estarem plenamente estáveis.

A Figura 34 apresenta um diagrama de tempo que permite visualizar a sequência de eventos tanto em operação normal como quando ocorre uma falha por SEU.

Figura 34 - Diagrama de tempo



Fonte: Autor

Nesta figura a operação normal é assim descrita: Quando ocorre um sinal de *clock* de escrita em CKW os controles *tri-state* Z1 e Z2 são imediatamente desligados e Z0 é ligado. Os *clocks* de cada registrador (CK1 e CK2) são atrasados pelos componentes DLY. Os resultados de cálculo de paridade P1 e P2 podem ou não exibir erro momentaneamente durante a escrita em R1 e R2 (marcas hachuradas na figura). Isso não afeta o funcionamento pois Z1 e Z2 estão já desligados pelo CKW. No final da escrita CKW retorna a zero e tanto Z1 como Z2 são liberados para exibir o dado escrito no barramento de saída.

No exemplo da figura um SEU ocorre no registrador R1 fazendo a paridade P1 acusar a falha (indicado por A na figura) e desligar Z1 (indicado por B na figura). Z2 permanece ativo e o barramento de saída não sofre nenhuma alteração. Eventualmente o sinal de *clock* CKsys é acionado causando a reescrita dos dados corretos em R1 (indicado por C na figura).

Caso o sinal DATA VALID esteja inativo (tanto R1 como R2 apresentam falha de paridade) nenhum registrador será gravado, pois com Z1 e Z2 desligados o barramento DATA OUT tem conteúdo desconhecido. No caso de DATA VALID ficar inativo o circuito será renovado apenas quando chegar um novo CKW.

Como os dados de saída estão presentes tão rápido quanto o atraso na passagem por Z0 este circuito pode ser utilizado em arquiteturas de alta velocidade. A limitação está em CKW que precisa ter largura suficiente para contabilizar os dois blocos de atrasos (DLY) mais o tempo de escrita do registrador R2. O circuito de cálculo de paridade deve ser mais rápido do que o primeiro bloco DLY.

O uso de portas lógicas na linha de *clock* dos registradores não é prática comum em projetos empregando FPGAs mas pode ser considerado em desenhos de circuitos integrados, e foi com este propósito que esta arquitetura em especial foi pensada. Isto porque os atrasos inerentes destas portas lógicas podem ser considerados no projeto com bastante confiabilidade. Além disso, a ocorrência de um SET nestes elementos do circuito não produziria falhas pois, causando um pulso na entrada de *clock* de um registrador, apenas memorizaria os dados já existentes nele (que voltam através do barramento DATA OUT).

O emprego de um habilitador (clock enable) neste circuito não é tarefa simples, visto que existem dois *clocks* que podem ser usados para excitar os registradores. Dentro da PSM este circuito empregaria apenas um sinal de *clock* defasado, porém sem o complemento do *clock enable* correspondente. Da forma como o circuito opera, a ocorrência de um SET na linha de *clock* em qualquer ponto deste circuito apenas resultaria num novo registro do mesmo dado que já está memorizado, sendo assim, inócuo para o funcionamento do sistema.

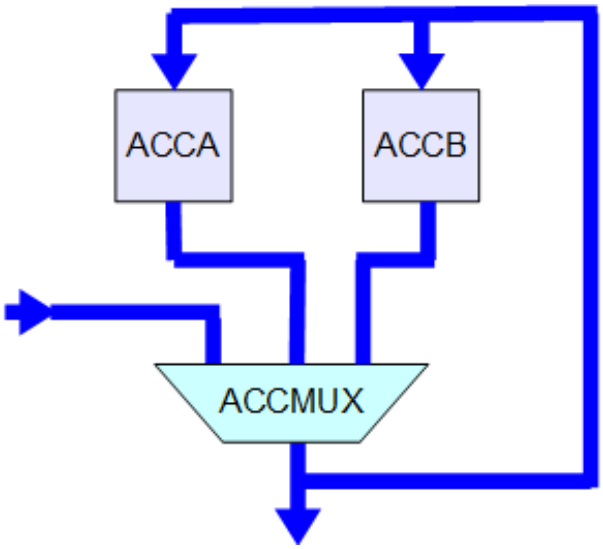
Este esquema de proteção pode ser implementado em bancos de registradores comumente utilizados em microprocessadores, permitindo que cada registrador mantenha sua integridade independente de uma supervisão de outros circuitos ou mesmo de um programa específico no microprocessador.

4.3 Registrador Auto Recuperável Empregando Seletor Único

Um outro desenho de circuito para construir um registrador de dados auto recuperável pode ser feito empregando um único seletor (multiplex) como mostrado na Figura 35.

Esta solução pode ser conveniente, apesar do aumento da quantidade de portas lógicas necessárias para montar o seletor de dados, porque evita a condição de contenção de barramento explicada anteriormente.

Figura 35 - Redundância de registrador com seletor único



Fonte: Autor

Neste circuito, quando um dado for escrito no Acumulador, o seletor ACCMUX vai transferir para a saída imediatamente e já estará disponível nas entradas dos registradores ACCA e ACCB, onde são escritos. Depois desta operação o ACCMUX seleciona o ACCA para sua saída e caso ocorra um SEU neste registrador vai selecionar o ACCB para manter a saída sempre correta.

O circuito contém um cálculo e verificação de paridade, atrasos de *clock* e indicador de dados válidos, como no item 4.2, porém não foram desenhados na Figura 35. A diferença entre os circuitos propostos está no uso de um seletor de dados (Figura 35) em vez de amplificadores *tri-state* (Figura 33).

Na eventualidade de um SEU ocorrer em ACCB, mesmo não causando erros no sistema porque o ACCA é normalmente o registrador selecionado, a lógica de recuperação faz reescrever o dado correto em ACCB, restaurando sua integridade.

Esta recuperação deve ocorrer antes que o dado armazenado em ACCA possa sofrer alguma perturbação por SEU, por isso a lógica que faz esta operação deve utilizar um sinal de *clock* apropriado.

4.4 Análise das arquiteturas de registradores auto recuperáveis

Na Tabela 3 é mostrada uma comparação da implementação das três arquiteturas de registrador auto recuperável aqui discutidas.

Tabela 5 - Recursos ocupados em cada implementação

Arquitetura	Qnt Flip-flops	Qnt FPGA Slices
PSM original com 3 seletores	18	24
Registrador auto recuperável empregando *tri-state*	18	17
Registrador auto recuperável empregando um só seletor	18	27

Fonte: Autor

A quantidade de *slices* indicada na tabela refere-se a uma unidade de ocupação de recursos de um FPGA Xilinx Spartan 3AN (o mesmo que foi utilizado no projeto do Controlador de Flaps). Uma implementação em Silício, com projeto dedicado do circuito integrado, deve resultar em números um pouco diferentes, porém uma comparação assim já demonstra a relação aproximada de ocupação de área em cada uma das arquiteturas.

A implementação proposta na PSM original exige poucos recursos, mas o barramento de saída só vai apresentar o dado sendo escrito depois que tiver ocorrido o *clock* de escrita nos registradores. A implementação com um só seletor (Figura 35) consome mais recursos, mas disponibiliza imediatamente o dado no barramento de saída.

A implementação mais econômica é aquela empregando *tri-state*, exigindo apenas 70% dos recursos da arquitetura original, porém, como visto anteriormente, pode causar pulsos de corrente quando ocorre um SEU. Esta implementação num FPGA não é tão eficiente quanto seria em Silício. Isso porque a maioria dos modelos de FPGAs possuem portas *tri-state* apenas junto aos pinos de saída. Nestes casos, a ferramenta de síntese pode implementar circuitos equivalentes (como os seletores de dados) e a quantidade de recursos empregada não reflete adequadamente a diferença entre as arquiteturas. Portanto, é razoável inferir que numa implementação em Silício a economia poder ser bem maior do que a calculada na Tabela 3.

5 CONCLUSÃO

Neste trabalho foram apresentados os conceitos sobre os efeitos da incidência de partículas de alta energia em dispositivos semicondutores, denominados SEE. A quantidade de energia depositada por estas partículas causa pulsos de corrente elétrica capazes de modificar o estado lógico de registradores e memórias, ou causar distúrbios em sinais críticos que podem mudar o resultado de operações e funções de máquinas de estado finitas.

Foram também apresentadas técnicas e métodos empregados para mitigar estes efeitos, ou mesmo detectar e resolver falhas causadas por estas partículas.

Objetivando o emprego de tecnologia de circuitos empregados em aplicações aeronáuticas, foram desenvolvidas algumas soluções para máquinas de estado finitas programáveis e circuitos de autocorreção para registradores de dados que são capazes de tolerar problemas causados por SEU e SET, corrigindo as falhas e mantendo o funcionamento correto do sistema.

Isso foi obtido através da aplicação de circuitos de verificação de dados por paridade, juntamente com a técnica de projetar uma máquina de estados de forma a existir apenas dois possíveis novos estados a cada instante. Todos os estados da PSM são guardados em uma memória insensível a falhas por SEU ou SET (do tipo ROM ou PROM). Foi possível demonstrar uma arquitetura digital para máquinas de estado programáveis que é tolerante a falhas por SEU e SET.

A tolerância obtida não evita que partículas energéticas alterem o estado de alguma memória ou registrador, porém o circuito desenvolvido permite recuperar o valor correto destes registros, mantendo a máquina de estados funcionando e com fidelidade nos dados tratados. Este funcionamento foi verificado. Testes demonstraram que falhas injetadas em todos os pontos sensíveis do circuito não causaram erros operacionais ou nos resultados do trabalho da máquina de estados.

Tal tolerância torna esta arquitetura favorável para utilização em sistemas que exigem confiabilidade e, por isso, foi empregada com sucesso em equipamentos aeronáuticos que foram fabricados e estão em uso atualmente.

Este estudo pode ser estendido através do projeto e implementação da arquitetura proposta com a fabricação de um circuito integrado. Uma implementação assim deveria contemplar circuitos de PSM convencionais e com a arquitetura proposta para poder oferecer meios de comparação da vulnerabilidade às ocorrências de SEE quando ambos circuitos são submetidos a radiação sob condições idênticas.

Testes num circuito integrado trariam ainda o benefício de produzir uma medida de comparação com componentes, disponíveis comercialmente, que executem as mesmas funções. Estes resultados poderiam qualificar o circuito proposto de forma acreditada e assim tornar-se um item comum para uso em aplicações que exijam alto grau de confiabilidade, obtendo mais facilmente a aceitação junto as entidades certificadoras de sistemas aeronáuticos e espaciais.

REFERÊNCIAS

ADELL, P.; SCHRIMPF, R. D.; BARNABY, H. J.; MAREC, R.; CHATRY, C.; CALVEL, P.; BARILLOT, C.; MION, O. Analysis of single-event transients in analog circuits. IEEE Transactions on Nuclear Science, v. 47, n. 6 III, p. 2616–2623, 2000.

ALIOTO, M.; CATALDO, G. DI; PALUMBO, G. Optimized design of high fan-in multiplexers using tri-state buffers. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, v. 49, n. 10, p. 1500–1505, 2002.

ANELLI, G. *et al.* Radiation Tolerant VLSI Circuits in Standard Deep Submicron CMOS Technologies for the LHC Experiments : Practical Design Aspects. IEEE Transactions on Nuclear Science, n. 46, p. 1690–1696, 2000.

ANGHEL, L.; ALEXANDRESCU, D.; NICOLAIDIS, M. Evaluation of a Soft Error Tolerance Technique Based on Time and / or Space Redundancy. 2000.

BALASUBRAMANIAN, A.; MEMBER, S.; NARASIMHAM, B.; MEMBER, S.; BHUVA, B. L.; MEMBER, S.; MASSENGILL, L. W.; EATON, P. H.; SIBLEY, M.; MAVIS, D. Implications of Total Dose on Single-Event Transient (SET) Pulse Width Measurement Techniques. v. 55, n. 6, p. 3336–3341, 2008.

BALASUBRAMANIAN, P.; PRASAD, K.; MASTORAKIS, N. E. A Fault Tolerance Improved Majority Voter for TMR System Architectures. v. 15, p. 108–122, 2016.

BAUMANN, R. C.; SMITH, E. B. Neutron-induced 10B fission as a major source of soft errors in high density SRAMs. Microelectronics Reliability, v. 41, n. 2, p. 211–218, 2001.

BAYLAKOGLU, I.; HUDAVERDI, M. Reliability Concerns of Radiation Effects on Space Electronics. 2012.

BRANCUS, I. M.; REBEL, H. Cosmic muons, as messengers from the Universe. AIP Conference Proceedings, v. 1645, n. February, p. 178–187, 2015.

BUCHER, M.; NIKOLAOU, A.; PAPADOPOULOU, A.; MAKRIS, N.; CHEVAS, L.; BORGHELLO, G.; KOCH, H. D.; FACCIO, F. Total Ionizing Dose Effects on Analog Performance of 65 nm Bulk CMOS with Enclosed-Gate and Standard Layout. 2018 IEEE International Conference on Microelectronic Test Structures (ICMTS), p. 166–170, 2018.

CASSEL, M.; KASTENSMIDT, F. L. Evaluating One-Hot Encoding Finite State Machines for SEU Reliability in SRAM-based FPGAs. 2006.

CAST. CAST-30 Simple Electronic Hardware. [s.l: s.n.].

CAZEAUX, J. M.; OMA, M.; NW, A. D. On Transistor Level Gate Sizing for Increased Robustness to Transient Faults ∗. 2005.

CHERTKOW, O.; PESCOVSKY, A.; ATIAS, L.; FISH, A. A Novel Low Power Bitcell Design Featuring Inherent SEU Prevention and Self Correction Capabilities. J. Low Power Electron, p. 130–150, 2015.

DETCHEVERRY, C.; DACHS, C.; LORFEVRE, E.; SUDRE, C.; BRUGUIER, G.; GASIOT, J.; ECOFFET, R.; MONTPELLIER, U. SEU |Critical Charge and Sensitive Area in a Submicron CMOS Technology. v. 44, n. 6, 1997.

DJORDJEVIC, G. L.; STANKOVIC, T. R.; STOJCEV, M. K. Approach to partially self-checking finite state machine design. n. July, 2014.

EASA. Regular update of AMC-20 : AMC 20-152 on Airborne Electronic Hardware and AMC 20-189 on Management of Open Problem Reports. [s.l: s.n.].

EINSFELDT, A.; GIACOMINI, R. Fault-tolerant architecture with full recovery under presence of SEU2018 IEEE 19th Latin-American Test Symposium, LATS 2018. Anais...2018

EINSFELDT, A.; GIACOMINI, R. Self-Recoverable Data Register Under Presence of SEESEMINATEC 2018. Anais...2018

EL-MALEH, A. H.; AL-QAHTANI, A. S. A finite state machine based fault tolerance technique for sequential circuits. Microelectronics Reliability, v. 54, n. 3, p. 654–661, 2014.

ERNST, D.; KIM, N. S.; DAS, S.; PANT, S.; RAO, R.; PHAM, T.; ZIESLER, C.; BLAAUW, D.; AUSTIN, T.; FLAUTNER, K.; MUDGE, T. Razor: A low-power pipeline based on circuit-level timing speculation. Proceedings of the Annual International Symposium on Microarchitecture, MICRO, v. 2003–Janua, p. 7–18, 2003.

FAA. AC23.1309. [s.l: s.n.].

FACCIO, F. Radiation effects in the electronics for CMS. 2000.

FACCIO, F. Large Scale Application of Design Hardening, 2005.

FEDERICO, C. A.; GONÇALEZ, O. L.; FONSECA, E. S.; MARTIN, I. M.; CALDAS, L. V. E. Neutron spectra measurements in the south Atlantic anomaly region. Radiation Measurements, v. 45, n. 10, p. 1526–1528, 2010.

FRIGERIO, L.; SALICE, F. RAM-based fault tolerant state machines for FPGAs. Proceedings - IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, p. 312–320, 2007.

FURUTA, J.; KOBAYASHI, K.; ONODERA, H. An Area / Delay Efficient Dual-Modular Flip-Flop with Higher SEU / SET Immunity. n. 3, p. 340–346, 2010.

GAILLARD, R. Single Event Effects: Mechanisms and Classification. [s.l: s.n.].

GARCIA-VARGAS, I.; SENHADJI-NAVARRO, R.; JIMENEZ-MORENO, G.; CIVIT-BALCELLS, A.; GUERRA-GUTIERREZ, P. ROM-Based Finite State Machine Implementation in Low Cost FPGAs. 2007 IEEE International Symposium on Industrial Electronics, n. 12, p. 2342–2347, 2007.

GARVIE, M.; HUSBANDS, P. Automatic Synthesis of Totally Self-Checking Circuits. n. January, 2019.

HADJICOSTIS, C. N.; VERGHESE, G. C. Coding Approaches to Fault Tolerance in Linear Dynamic Systems. v. 51, n. 1, p. 210–228, 2005.

HAMMETT, R.; CHARLES, T.; DRAPER, S. DESIGN BY EXTRAPOLATION AN EVALUATION OF FAULT-TOLERANT AVIONICS Fault-Tolerant System Requirements. p. 1–12, 2001.

HASANBEGOVIC, A.; AUNET, S. Heavy Ion Characterization of Temporal-, Dual- and Triple Redundant Flip-Flops Across a Wide Supply Voltage Range in a 65 nm Bulk CMOS Process. IEEE Transactions on Nuclear Science, v. 63, n. 6, p. 2962–2970, 2016.

HOLBERT, K. Single Event Effects. http://holbert.faculty.asu.edu/eee560/see.html, 2006.

HUANG, Z.; LIANG, H. The Impact of MBUs on the Reliability of Rollback Recovery Circuits. 2009.

JOSHI, B.; PRADHAN, D. K.; MOHANTY, S. P. Fault Tolerant Nanocomputing. 2010.

KAMAROU, V. D. A. Radiation effects and damage formation in semiconductors due to high-energy ion irradiation. [s.l: s.n.].

KASTENSMIDT, F. L.; STERPONE, L.; CARRO, L.; REORDA, M. S. On the Optimal Design of Triple Modular Redundancy Logic for SRAM-based FPGAs. Design, Automation and Test in Europe, p. 1290–1295, 2005.

KIM, E. P.; SHANBHAG, N. R. Soft N-Modular Redundancy. p. 1–16, 2010.

KIM, K. S.; MITRA, S.; SEIFERT, N.; QUAN SHI, M. Z. Robust System Design with Built-In Soft-Error Resilience. n. February, p. 43–52, 2005.

KOGA, R.; YU, P.; CRAWFORD, K.; GEORGE, J.; ZAKRZEWSKI, M. Synergistic Effects of Total Ionizing Dose on SEU Sensitive SRAMs. v. 36, p. 1–6, 2009.

KOOPMAN, P.; DRISCOLL, K. Tutorial : Checksum and CRC Data Integrity. 2012.

KOREN, I.; SU, S. Reliability Analysis of N-Modular Redundancy Systems with Intermittent and Permanent Faults. v. c, n. 7, p. 514–520, 1979.

KUMAR, S. Fundamental Limits to Moore ' s Law. n. December, 2015.

KUMAR, S.; AGARWAL, S.; JUNG, J. P. Soft error issue and importance of low alpha solders for microelectronics packaging. Reviews on Advanced Materials Science, v. 34, n. 2, p. 185–202, 2013.

KUMAR, S.; CHELLAPPA, S.; CLARK, L. T. Temporal pulse-clocked multi-bit flip-flop mitigating SET and SEUProceedings - IEEE International Symposium on Circuits and Systems. Anais...2015

LABEL, K. A. Single Event Effect Criticality Analysis. p. 3–5, 1996.

LAMPORT, L.; SHOSTAK, R.; PEASE, M. The Byzantine Generals Problem. v. 4, n. 3, p. 382–401, 1982.

LAYTON, P.; LONGDEN, L.; PATNAUDE, E. Evaluation of Reliability and Data Retention of an Irradiated Nonvolatile Memory. p. 1–5, 2005.

LI, T.; YANG, H.; CAI, G.; ZHI, T.; LI, Y. A CMOS Triple Inter-Locked Latch for SEU Insensitivity Design. v. 61, n. 6, p. 3265–3273, 2014.

LIN, Y.; ZWOLINSKI, M. SETTOFF: A fault tolerant flip-flop for building Cost-efficient Reliable Systems. Proceedings of the 2012 IEEE 18th International On-Line Testing Symposium, IOLTS 2012, p. 7–12, 2012.

LUBA, T.; GORSKI, K.; WRONSKI, L. B. ROM-based finite state machines with PLA address modifiers. Proceedings EURO-DAC '92: European Design Automation Conference, p. 272–277, 1992.

MAY, T. C.; WOODS, M. H. Alpha-particle-induced soft errors in dynamic memories. IEEE Transactions on Electron Devices, v. 26, n. 1, p. 2–9, 14 jan. 1979.

MCCLUSKEY, E.; WAKERLY, J.; OGUS, R. Center for reliable computing current research No.100. [s.l: s.n.].

MCCOLLUM, J. *et al.* Current radiation issues for programmable elements and devices. IEEE Transactions on Nuclear Science, v. 45, n. 6, p. 2600–2610, 2002.

MESSENGER, G. C.; SPRATT, J. P. The Effects of Neutron Irradiation on Germanium and SiliconIRE proceedings. Anais...1958

MILLER, F.; LUU, A.; PRUD'HOMME, F.; POIROT, P.; GAILLARD, R.; BUARD, N.; CARRIÈRE, T. Characterization of single-event burnout in power MOSFET using backside laser testing. IEEE Transactions on Nuclear Science, v. 53, n. 6, p. 3145–3152, 2006.

MOHANRAM, K. Simulation of transients caused by single-event upsets in combinational logic. Proceedings - International Test Conference, v. 2005, p. 973–981, 2005.

MULAZZANI, M. RELIABILITY VERSUS SAFETY. IFAC Proceedings Volumes, v. 18, n. 12, p. 141–146, 1985.

NIDHIN, T. S.; BHATTACHARYYA, A.; BEHERA, R. P.; JAYANTHI, T.; VELUSAMY, K. Verification of fault tolerant techniques in finite state machines using simulation based fault injection targeted at FPGAs for SEU mitigation. Proceedings of 2017 4th International Conference on Electronics and Communication Systems, ICECS 2017, v. 17, 2017.

PILTAN, F.; AVATEFIPOUR, O.; SOLTANI, S.; MAHMOUDI, O. Design FPGA-Based CL-Minimum Control Unit. n. January, 2016.

POIVEY, C. TNID Total Non Ionizing Dose or DD Displacement Damage. ESA - CERN - SCC Workshop, 2017.

REZGUI, S.; MCCOLLUM, J.; WON, R. Design and Layout Effects on SET Propagation in 90-nm ASIC and FPGA Test Structures. v. 57, n. 6, p. 3716–3724, 2010.

ROCKETT, L. An SEU-Hardned CMOS Data Latch Design. IEEE Transactions on Nuclear Science, v. 35, n. 6, p. 1682–1687, 1988.

SILVEIRA, M. A. G.; SANTOS, R. B. B.; LEITE, F. G. H.; ARAÚJO, N. E.; MEDINA, N. H.; PORCHER, B. C.; AGUIAR, V. A. P.; ADDED, N.; VARGAS, F.; IEEE, M. X-ray-induced Upsets in a Xilinx Spartan 3E FPGA. 2015.

SNOEYS, W. J.; GUTIERREZ, T. A. P.; ANELLI, G. A New NMOS Layout Structure for Radiation Tolerance. n. September, 2002.

SOUZA, R. N.; SILVEIRA, M. A. G.; GIMENEZ, S. P. Improving MOSFETs radiation robustness by using the wave layout to boost analog ICs applications. 2014 29th Symposium on Microelectronics Technology and Devices: Chip in Aracaju, SBMicro 2014, p. 30–37, 2014.

SPIELER, H. Introduction to radiation-resistant semiconductor devices and circuits. AIP Conference Proceedings, v. 390, p. 23–49, 1997.

SROUR, J. R.; MARSHALL, C. J.; MARSHALL, P. W. Review of Displacement Dmage Effects in Silicon Devices. v. 50, n. 3, p. 653–670, 2003.

THOMAS, E. Certification Cost Estimates for Future Communication Radio Platforms. [s.l: s.n.].

TODD, B.; UZNANSKI, S. Radiation Risks and Mitigation in Electronic Systems. v. 003, n.

May 2014, p. 7–14, 2015.

TORELL, W.; AVELAR, V. Mean Time Between Failure : Explanation and Standards. n. January 2004, 2004.

TRAINOR, J. H. INSTRUMENT AND SPACECRAFT FAULTS ASSOCIATED WITH NUCLEAR RADIATION IN SPACE. v. 14, n. 10, p. 685–693, 1994.

TRIVEDI, R.; MEHTA, U. S. A survey of radiation hardening by design (rhbd) techniques for electronic systems for space application. International Journal of Electronics and Communication Engineering & Technology (IJECET), v. 7, n. 1, p. 75–86, 2016.

TRUYEN, D.; BOCH, J.; SAGNES, B.; RENAUD, N.; LEDUC, E.; ARNAL, S.; SAIGNÉ, F. Temperature Effect on Heavy-Ion Induced Parasitic Current on SRAM by Device Simulation : Effect on SEU Sensitivity. v. 54, n. 4, p. 1025–1029, 2007.

VAIDYA, B.; ANUPAM, D. A Study of different Finite state machine design and efficient coding techniques. p. 493–496, 2013.

VILLA, P.; BEZERRA, E.; MEDINA, N.; ADDED, N.; AGUIAR, V. DE; MACCHIONE, E.; AGUIRRE, F.; PAULO, S. Analysis of COTS FPGA SEU-Sensitivity to Combined Effects of Conducted-EMI and TID. p. 27–32, 2017.

WAKERLY, J. Reliability of Microcomputer Systems Using Triple Modular Redundancy, 1975.

WANG, F.; AGRAWAL, V. D. Single event upset: An embedded tutorial. Proceedings of the IEEE International Frequency Control Symposium and Exposition, p. 429–434, 2008.

WANG, F.; AGRAWAL, V. D. Single Event Upset : An Embedded Tutorial. p. 429–434, 2008.

WIRTH, G.; KASTENSMIDT, F. L.; RIBEIRO, I. Single Event Transients in Logic Circuits — Load and Propagation Induced Pulse Broadening. v. 55, n. 6, p. 2928–2935, 2008.

XAPSOS, M. A.; STAUFFER, C.; PHAN, A.; MCCLURE, S. S.; LADBURY, R. L.; PELLISH, J. A.; CAMPOLA, M. J.; LABEL, K. A. CONFIDENCE LEVEL BASED APPROACH TO TOTAL DOSE SPECIFICATION FOR SPACECRAFT ELECTRONICS. 2017.

YAMADA, T.; ABE, M.; NITTA, Y.; OGURA, K.; KUSAOKE, M. Reducing Consuming Clock Power Optimization of a 90 nm Embedded Core. n. March 2006, 2006.

YOUNGQUIST, R. C.; NURGE, M. A.; STARR, S. O.; KOONTZ, S. L. Thick galactic cosmic radiation shielding using atmospheric data. Acta Astronautica, v. 94, n. 1, p. 132–138, 2014.

YUAN, L.; ZHAO-WEI, S.; DAN, Z. Research on the Problems of Satellite Borne FPGA Based Finite State Machine. p. 0–3, 2008.

ZENG, C.; SAXENA, N.; MCCLUSKEY, E. J. Finite state machine synthesis with concurrent error detection. p. 672–679, 1999.

ZIEGLER, J. F. Terrestrial Cosmic Rays. IBM Reserach & Development, v. 40, n. 1, p. 19–39, 1996.

APÊNDICE A – TRABALHOS PUBLICADOS

Durante os estudos desenvolvidos neste curso de Mestrado foram apresentados dois trabalhos em eventos nacionais e internacionais, como segue:

Fault-tolerant architecture with full recovery under presence of SEU, apresentado no IEEE 19th Latin-American Test Symposium, LATS 2018.

Self-Recoverable Data Register Under Presence of SEE, apresentado no SEMINATEC 2018.

Fault-Tolerant Architecture With Full Recovery Under Presence of SEU

Augusto Einsfeldt
Department of Electrical Engineering
FEI
São Bernardo do Campo, Brazil
aee@terra.com.br

Renato Giacomini
Department of Electrical Engineering
FEI
São Bernardo do Campo, Brazil
renato@fei.edu.br

***Abstract*—A SEU fault-tolerant finite-state machine architecture is presented. It does use less resources than triple redundancy checking techniques and performs a verification to confirm that each operation has been completed without errors, before allowing to step further. A direct addressing technique is used to reduce the risk to advance to an unknown state due to some upsetting event between states. The complete architecture was implemented in a FPGA, including worst-case fault injection structure, to evaluate the reliability. Results have shown effective fault-tolerant behavior and no data loss.**

Keywords—finite-state machine; SEU; fault-tolerant; reliability; recovering

I. Introduction

A fault-tolerant circuit shall remain reliable in the presence of high-energy particle strikes that could cause a SEE (Single Event Effect). A SEU (Single Event Upset) and a SET (Single Event Transient)[1][2], also known as soft errors, are types of SEE and means the error introduced by radiation effect does not affect the hardware structure and can be recoverable. There are several types of SEE[3] that are not recoverable like SEL (Single Event Latch-up), SEB (Single Event Burnout) and even cumulative effects like TID (Total Ionizing Dose) or DD (Displacement Damage) and the solution for those won't be addressed in this work.

A reliable hardware shall exhibit ways to prevent SEE or to work correctly upon the event. The most sensitive hardware is a flip-flop. This storage element holds a bit of information in a discrete system and a SEE can affect its contents causing a transient in the clock line, in the data line when a clock event is present or by changing the internal state of the stored bit.

There are several ways to handle soft errors and a triple modular redundant (TMR)[4] circuit is a common solution for reliable systems. Since TMR uses more silicon and power with potential reduction in performance many researchers have been working to find better solutions. This includes RHBD[3] techniques like RAZOR[5], EDC[6] or reversible gates[7].

This work proposes a new architecture concept for a reliable programmable state machine (PSM) that is tolerant to SEU and SET. A PSM has been chosen to demonstrate the concept due to its versatility. It does use parity checking as means to detect SEE and logic structures to solve it. Similar to other PSM approaches it uses part of the memory's data word to address the next state along with input values[8][9]. The proposed architecture provides full fault-tolerance to ongoing SEE using fewer resources than TMR while presenting a highly reliable data processing design, useful for critical applications.

II. Description

This programmable state machine uses a ROM with embedded parity to store the possible outputs for each state and the address for the next state. It is shown in Figure 1. It presents a structure similar to a conventional microcontroller but without an instruction counter. The parity checking and fault resolution circuits results in a TSC (totally self-checking) type of sequential circuit[10]. Different PSM designs could be implemented using the same fault-tolerant techniques described in this work.

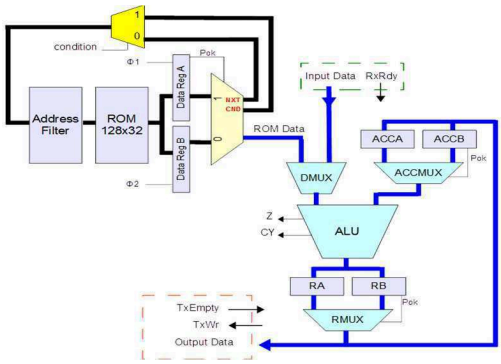


Figure 1: Complete state machine

It is assumed the ROM content is not modifiable by a SEE, although some commercially available EEPROM and FLASH memories can be sensitive to high radiation doses[11]. A reliable ROM can be built in silicon (not allowing changes in program code) or with anti-fused OTP memory, allowing on-field programmability.

Different from conventional ROM-based PSM where the next state address is a combination of current state ROM's data and input values[9] the proposed architecture has only two possible next state addresses: one when the internal condition is true and one when the internal condition is false (Figure 2).

978-1-5386-1472-3/18/$31.00 ©2018 IEEE

ROM 32 bit data word			
31 – 24	23 – 16	15 – 8	7 – 0
control lines	8 bit data	next address if condition true	next address if condition false

Figure 2: Contents of every ROM position

The internal condition signal comes from status lines like the zero-result value in the ALU. The address bus in this test architecture is 7 bit wide and data bus is 32 bits wide. The address value stored in the ROM's data word uses 8 bits (7 bits plus an embedded parity bit). Figure 3 shows the address path.

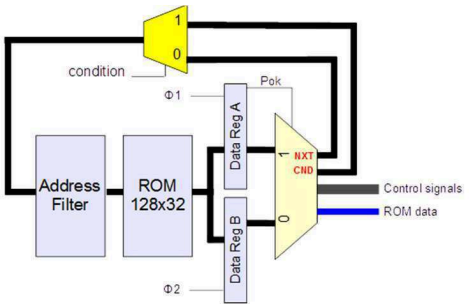


Figure 3: Address path

Two registers will hold ROM's data word. Each has a clock in a different phase. If a SET happens in one of ROM's data lines it may be stored in one registers but not in both, leading to a detectable parity fault. Also, if one bit is flipped due to a SEU the parity error will command the multiplex to select the other register that shall be correct. Note that this architecture is valid for a SEE and not for multiple simultaneous events. Two flipped bits in one register may result in a non-detectable fault. Due to inherent criticality of clock lines to a SET the flip-flop structure (Figure 5) uses an AND gate for clock enabling. The gate's inputs shall come from two separated line drivers. A SET striking one input would not be carried to the output. A SET striking the gate's output could enable one flip-flop of the register, only, potentially causing a detectable bit flip.

The address filter is a mechanism to solve a SET in the condition line. In this test implementation it uses 3 registers with a majority vote. A silicon implementation could consider the memory access time to absorb the 1ns transient expected from a SET[2], eliminating the need for a filter.

The system's clock could be supplied as eight separate clocks with delay between each other or a single global clock with eight clock enables (using a single clock domain). The delays must be larger than the duration of the longest expected transient from a particle strike. A 100MHz system's clock would use 8 clock pulses with 1.25ns from each other. Whit global clock it should be connected with two separated clock drivers in two lines and an AND gate at the flip-flop's clock input.

The data processing path is presented in the Figure 4.

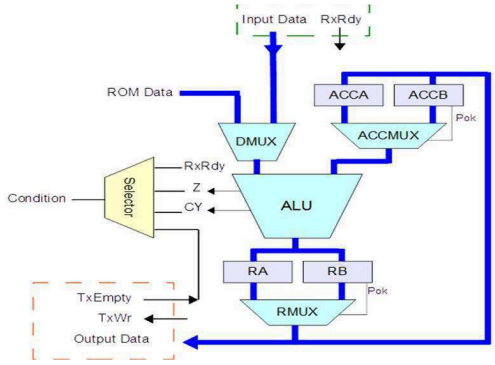


Figure 4: Data path

In this study few operations were implemented in the ALU: DMUX pass-through; DMUX XOR ACC; DMUX AND ACC; ACC minus DMUX; and Result Left Rotate. A Carry bit from subtraction operation and a Zero bit from result-equal-zero are used as condition for address selection. The result's data path uses two registers with clocks in different phases. If the RB register has been affected by a SEE its parity fault will command the RMUX to select the RA register. Same structure is used with the Accumulator (ACC).

Additionally, the RA and RB contents are checked to be equal after their phased clocks. A SET in the ALU logic may result in a valid parity calculation to be stored in RA or RB. To catch this failure, the registers must be compared. If they are different and presents a correct parity then the system must stall the clock till the next clocks for RA and RB will enter the correct values in these registers. This scheme is not necessary in ROM's data registers because the parity is embedded in the data word. Accumulator's registers also have the parity coming from RA or RB through the multiplex.

This architecture has the distinct advantage of using less resources than TMR. It can also be faster since phase (delay) displacement is an important strategy to avoid SET effects in TMR and TMR needs three clocks instead of two [12][13].

III. Reliability Verification

The verification method considers that the proposed architecture should recover from an ongoing SEE. The LET (Linear Energy Transfer) level is not relevant for recoverability verification. Therefore, the prevention of a SEE is not a concern. Also, all logic gates and flip-flops are conventional and uses no RHBD[3] techniques to prevent bit flip or transient propagation.

Each flip-flop has the structure shown in Figure 5. This structure is suited for implementation in a FPGA with a global clock with clock enables. When a fault is to be injected the Fault_enable signal must be active along with the Fault_bit. Then the next clock edge will replace the flip-flop's content with opposite value it had before, emulating a SEU that could flip the data or a SET in the data line during a clock event. The Fault_enable signal, during the reliability verification, will be active for one of the eight clock slots available on a given system's clock cycle. Since it will be performed on each clock slot and along the whole test duration this process guarantees a flip-flop upset on every possible time slice.

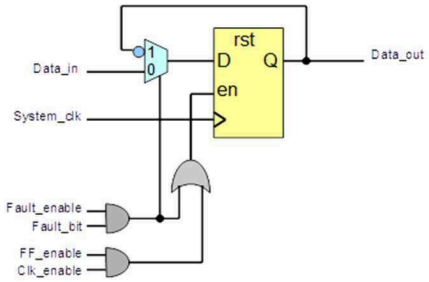


Figure 5: Flip-flop with Fault Injection

Tests were performed using two implementations of the state machine architecture: one as proposed fault-tolerant architecture and the other without the fault-tolerant techniques. When no fault is injected both implementations reported no failures and therefore served as baseline for fault responses. The proposed architecture's test implementation requires 120 fault injection signals and the not fault-tolerant specimen has 53 signals. The difference is because the late has no duplicated registers.

Both implementations run the same software and presents the same expected results when no fault happens. The software is a process with 12 stages. Each stage uses the result from the previous stage and produces a unique output. The first stage uses a data byte sent from a computer controlling the test.

When the PSM receives a data value 0x02 each stage will output, in sequence, the values: 0x3D, 0xB9, 0x87, 0x2F, 0x61, 0xA8, 0x73, 0xE0, 0xA2, 0xBB, 0x9F and the final result sent back to computer will be 0x0F. All stages are copies of the same algorithm that performs arithmetic operations, conditional jumps and so forth. The algorithm was chosen to result in a different value for each stage. Therefore, if a fault causes a wrong jump during the process the data returned to the computer would be different than the expected. This technique allows detection of bit faults along the whole program duration and memory positions.

A. Test Platform

The proposed architecture can be tested using electronic emulation with the circuit implemented in a FPGA and including a fault injection hardware that mimics the effects of particle strike in control signals and flip-flops. Valid results are obtained when injecting faults on all critical signals with proper timing. For instance, a SET in a signal before or after a clock event would cause no fault and doesn't need to be tested. This fault injection method would represent the worst-case scenario when every sensitive signal would be hit by a high energy particle during the process execution time. Real-life tests could result in less faults since many particles would hit signals when they are not sensitive. The presented test shall evaluate the fault-tolerance to SEE and not the probability of it to happen.

The test platform used a Xilinx Sparta-3A FPGA demo board. The FPGA contains the AUT (Architecture Under Test) circuit, the fault injection logic, the communication interface and a state machine to control the process. The FPGA's vendor and model are not relevant. The architecture was designed in VHDL and could be implemented in any FPGA for this evaluation.

B. Test State Machine

The Test State Machine controls the whole testing process. It produces the AUT's system clock, the fault injection signals and handles the communication with the computer. It does receive a 20 bytes long data string from computer, containing the fault injection pattern, the fault clock counter and the initial data byte to the AUT. Once receiving the string and checking its integrity the state machine will write the initial byte in the AUT's input bus and release the system clock.

Following this action, it will monitor the AUT for program completion. If the AUT never finish the program the state machine will exit after a timeout period. By timeout or noticing program completion it will stop the AUT's and send the results to the computer. An incorrect or lack of result means the injected fault has caused an unrecoverable fault. After each run the computer sends another string and the state machine restarts the AUT fresh to check reliability with a new fault injection set.

C. Test Method

The string sent to the Test Platform contains a fault clock count value. It means the fault will be injected when the AUT's has run that number of clock cycles (including the eight clock enables per system's clock). The test architecture used in this work uses up to 90 clocks to compute all 12 stages. With 8 clock phases it means 720 fault injection moments. This method allows injecting faults at any clock event along the whole program execution, covering every possible fault that could be injected. Counting all 720 clock events, all flip-flops and sensitive control signals results in 86,400 fault injections (720 x 120) for the proposed architecture and 38,160 (720 x 53) for the non fault-tolerant architecture. With this approach the test is conclusive with a SEE happening at every sensitive signal.

D. Results

The results are presented in the Table I.

TABLE I. TEST RESULTS

Architecture	Injection type	Injected faults	Failures	Failure Rate
fault-tolerant	Scan with every possible fault	86400	0	-
Non fault-tolerant	Scan with every possible fault	38160	10865	28%
non fault-tolerant	Random	82926	23920	29%

The non fault-tolerant architecture failed in 28% of the tests, meaning some injected fault did not result in failure. It does happen when an injected fault occurs in a flip-flop or control line that does not cause any harm to the process. A random test was performed to allow comparison with the effects caused by a full scan.

A double SEE injection was also performed in the fault-tolerant implementation to check the immunity in this mode. Since it was not designed for simultaneous events the number of failures was not satisfactory, resulting in an error rate of approximately 3.8% (1814 failures in 48104 of double events) with random injection.

E. Architecture resource usage comparison

The proposed architecture demanded 89% more resources over the resources used by the non fault-tolerant design. It is better than a TMR approach that usually demands more than 200% of resource increase (triple copy of registers plus voting

logic and delay elements). Table II shows the values from the architectures implemented in the test platform. The quantity of resources could vary when using a different FPGA or vendor but the magnitudes would remain similar.

TABLE II. TEST RESULTS

FPGA implementation	No of FFs	Total LUTs for logic	Additional FFs	Additional LUTs
Test structure only	175	241	-	-
non-fault tolerant	307	494	132	253
fault-tolerant	433	714	258	473

IV. Conclusion

The presented SEU fault-tolerant architecture exhibits a very effective way of implementing a reliable programmable state machine. It is fully fault-tolerant and high reliable on the presence of single events caused by high energy particles. When used with RHBD techniques and Hamming code could result in designs also tolerant to simultaneous events. The proposed architecture can keep the programmable state machine working with no errors during a SEE without the cost of TMR and using standard IC libraries.

Due to its high reliability this architecture can be used in mission critical applications. Application examples would be firing explosives or thrusters in rockets and supervising computers or other processes for behavior fault.

V. Future Work

The design can be further enhanced to improve tolerance to simultaneous upset events using Hamming code or even ECC instead of parity along with some RHBD techniques. The number of internal delayed clocks can be reduced using operation overlaps like storing Accumulator data at same time the ROM data is read to the registers. This would allow a faster system clock.

A silicon implementation could be developed and fabricated for testing under presence of high-energy particles.

VI. References

[1] F. Wang and V. D. Agrawal, "Single event upset: An embedded tutorial," *Proc. IEEE Int. Freq. Control Symp. Expo.*, pp. 429–434, 2008.

[2] K. Mohanram, "Simulation of transients caused by single-event upsets in combinational logic," *Proc. - Int. Test Conf.*, vol. 2005, pp. 973–981, 2005.

[3] R. Trivedi and U. S. Mehta, "A survey of radiation hardening by design (rhbd) techniques for electronic systems for space application," *Int. J. Electron. Commun. Eng. Technol.*, vol. 7, no. 1, pp. 75–86, 2016.

[4] F. L. Kastensmidt, L. Sterpone, L. Carro, and M. S. Reorda, "On the Optimal Design of Triple Modular Redundancy Logic for SRAM-based FPGAs," *Des. Autom. Test Eur.*, pp. 1290–1295, 2005.

[5] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," *Proc. Annu. Int. Symp. Microarchitecture, MICRO*, vol. 2003–Janua, pp. 7–18, 2003.

[6] N. Julai, A. Yakovlev, and A. Bystrov, "Error detection and correction of single event upset (SEU) tolerant latch," 2012, p. 6.

[7] T. Ravi, S. Ranjith, and V. Kannan, "A Novel Design of D-Flip Flop Using New RR Fault Tolerant Reversible Logic Gate," vol. 3, no. 2, 2013.

[8] T. Luba, K. Gorski, and L. B. Wronski, "ROM-based finite state machines with PLA address modifiers," *Proc. EURO-DAC '92 Eur. Des. Autom. Conf.*, pp. 272–277, 1992.

[9] I. Garcia-Vargas, R. Senhadji-Navarro, G. Jimenez-Moreno, A. Civit-Balcells, and P. Guerra-Gutierrez, "ROM-Based Finite State Machine Implementation in Low Cost FPGAs," *2007 IEEE Int. Symp. Ind. Electron.*, no. 12, pp. 2342–2347, 2007.

[10] C. Zeng, N. Saxena, and E. J. McCluskey, "Finite state machine synthesis with concurrent error detection," pp. 672–679, 1999.

[11] P. Layton, L. Longden, and E. Patnaude, "Evaluation of Reliability and Data Retention of an Irradiated Nonvolatile Memory," pp. 1–5, 2005.

[12] P. Mongkolkachit and B. Bhuva, "Design technique for mitigation of alpha-particle-induced single-event transients in combinational logic," *IEEE Trans. Device Mater. Reliab.*, vol. 3, no. 3, pp. 89–92, 2003.

[13] S. Kumar, S. Chellappa, and L. T. Clark, "Temporal pulse-clocked multi-bit flip-flop mitigating SET and SEU," in *Proceedings - IEEE International Symposium on Circuits and Systems*, 2015, vol. 2015–July, pp. 814–817.

Self-Recoverable Data Register Under Presence of SEE

Augusto Einsfeldt[1] and Renato Giacomini[1]

[1] Department of Electrical Engineering, FEI, São Bernardo do Campo, Brazil

e-mail: aee@terra.com.br, renato@fei.edu.br

A. 1. Abstract

The proposed data register design can keep the stored data valid after a single event effect (SEE) using integrity checking and a register clone. The data path uses tri-state buffers to accommodate a complex multiplexer and reduce the amount of silicon necessary for the implementation.

This data register structure could be expanded to build a register bank used in microprocessors improving the system reliability.

B. 2. Introduction

In applications from commercial devices to satellites and interplanetary probes the environment can exhibit high energy particles that can strike sensitive nodes in digital circuits and cause upsets by changing memory and flip-flop contents or causing clock events in an unexpected moment[1]. A reliable circuit withstands such events with shielding or recovering from it. An usual solution for data registers would be a Triple Modular Redundancy (TMR)[2] but it demands more silicon area and the majority voter is inherently difficult to design to reduce the sensitivity to SEE.

The proposed design, shown in Fig 1, duplicates the data register to have two copies of same content and provide means of recoverability. A parity bit is implemented as means of data integrity checking and is produced using a multiple input XOR gate. The three tri-state buffers, Z0, Z1 and Z2, form an economic multiplex circuit allowing the correct data content always to be present at Data Out bus. This approach of using tri-state buffers to form multiplexers has been used successfully in many designs[3]. When one bit of a data register is changed due to a SEE the fault is detected by parity checking and it will disable the tri-state buffer in use, while the correct data, from the other register, remains present at Data Out.

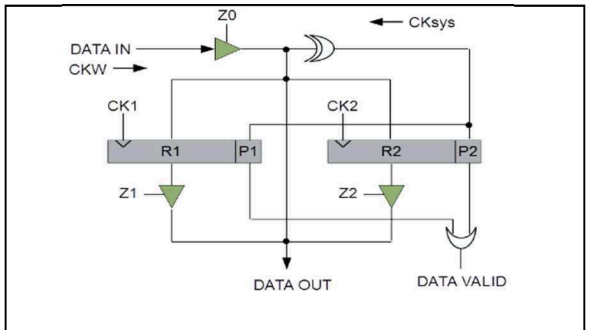


Fig.1. Simplified structure of recoverable Data Register.

Since the correct data at Data Out is also present at both register's inputs a clock event can re-write the faulty register with correct data, recovering it from fault.

The write clock input (CKW) provides the normal timing for the circuit. A group of delay devices (not shown in the figure) are used to provide adequate sequencing for register's clocks and tri-state buffers.

For instance, when CKW is active the circuit must enable the tri-state buffer Z0 and disable the others. The registers' clocks (CK1 and CK2) use delays to compensate the propagation time of the parity evaluation circuit. Since a SEE can strike at any signal an added reliability can be obtained if CK2 is at least 1ns apart from CK1 to produce a timing displacement[4]. P1 and P2 are parity checking results using a XOR with all register's bits plus the stored parity. While Z0 is active during CKW the Data Out will also presents the input value. When CKW is released, the parity checking P1 of R1 shall drive the tri-state buffer Z1. CKW width must be enough to have both registers written and their parity bits checked before CKW is released. This would avoid some race condition and possible bus contention when Z1 and Z2 could be enable before R1 and R2 contents have the same value. The logic governing the Z1 buffer is depicted in Fig 2. Z2 buffer has the same logic.

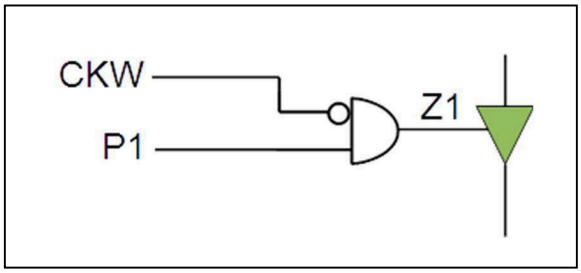


Fig.2. *Z1 Tri-State buffer logic.*

The logic will enable the Z1 buffer when CKW is not present and when P1is active, meaning the R1 data content is valid. The same will happen with Z2 and R2. Since both registers contains the same value then both tri-state buffers can be active with no harm to the system. In case R1 or R2 is changed due to a SEE the respective tri-state buffer will be disabled and the other register's buffer will remain active. There will be a momentary single bit condition with opposite driving,

but it shall last for two logic gate delays, only, till the faulty register's buffer is disabled.

A clock selector (an OR gate with an AND gate) placed at CK1's and CK2's clock inputs will allow a different external clock signal (CKsys) to re-write the faulty register. This clock can be generated by the system at any moment apart from CKW, although would be better if happening continuously, with short period, to reduce the probability of some SEE striking the currently healthy register before curing the other register.

In the event of multiple upsets causing failures in both registers a Data Valid output would be deactivated, signalling other circuits that the Data Out bus content is invalid till next CKW event.

C. 3. Timing

Fig 3 shows the timing diagram for this design. The first phase shows the circuit behavior when a new data is to be written. P1 and P2 may or may not be momentary inactive due to bit propagation delays. During this settling time the Data Out will remain valid since contains the Z0 buffer's output while Z1 and Z2 are disabled.

When a fault is detected (red circle in the figure) and once CKW is inactive the CKsys clock is allowed to generate a new CK1 clock and therefore re-write R1 with R2's contents. Since CKsys can cause a CK1 or CK2 clock event only when CKW is inactive, and P1 or P2 is also inactive, this means CKsys can be a high frequency and cyclic clock signal.

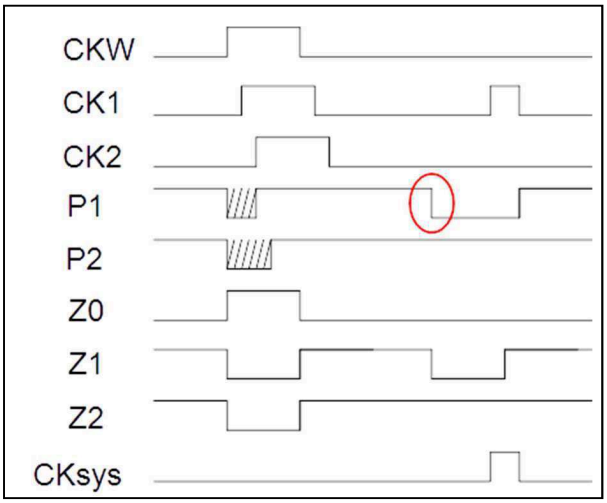


Fig.3. *Timing diagram.*

The delay mentioned before is seen with the displacement from CKW to CK1 and CK2.

Reliability under SEE is achieved by design. If a SET (Single Event Transient) in CK1 or CK2 happens in an unexpected moment it would only re-write the same contents in the registers. A SET in the CKW line would be absorbed by delay devices since a SET duration is known to be no longer than 1ns[4] and the delays can be elaborated to last a little longer. A parity error induced by SEU (Single Event Upset) would result in correct data and parity re-write and therefore be innocuous. A SET in one parity line, P1 or P2, would disable momentary one buffer but the other would keep the Data Out active with correct data.

The delay between CK1 and CK2 is also intended to absorb SET in the line between these signals[5], while providing sequencing and system's power reduction (less signals switching simultaneously).

D. 4. Conclusions

The presented design shows a solution for data register reliability under SEE with few logic gates and simple register duplication. Parity checking is obtained with simple XOR gate with multiple inputs and delay devices can be constructed with cascaded inverters or other logic gates.

The design allows fast logic. The CKW to Data Out delay is equal of Z0's buffer delay, only, and all other timings are transparent to external system. Therefore, this design can be used in GHz systems without compromising the performance. The limitation is CKW width must be wider than the register and parity circuit delays.

E. Acknowledgments

The authors thank the support from FEI and CAPES in granting the resources for this study.

F. References

[1] F. Wang and V. D. Agrawal, "Single event upset: An embedded tutorial," *Proc. IEEE Int. Freq. Control Symp. Expo.*, pp. 429–434, 2008.

[2] F. L. Kastensmidt, L. Sterpone, L. Carro, and M. S. Reorda, "On the Optimal Design of Triple Modular Redundancy Logic for SRAM-based FPGAs," *Des. Autom. Test Eur.*, pp. 1290–1295, 2005.

[3] M. Alioto, G. Di Cataldo, and G. Palumbo, "Optimized design of high fan-in multiplexers using tri-state buffers," *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, vol. 49, no. 10, pp. 1500–1505, 2002.

[4] K. Mohanram, "Simulation of transients caused by single-event upsets in combinational logic," *Proc. - Int. Test Conf.*, vol. 2005, pp. 973–981, 2005.

[5] S. Kumar, S. Chellappa, and L. T. Clark, "Temporal pulse-clocked multi-bit flip-flop mitigating SET and SEU," in *Proceedings - IEEE International Symposium on Circuits and Systems*, 2015, vol. 2015–July, pp. 814–817.