Desenvolvimento de Técnicas de Tolerância à Falhas para Componentes Programáveis por SRAM

Fernanda Gusmão de Lima Kastensmidt[1]

Gustavo Neuberger[2]

Luigi Carro[2]

Ricardo Reis[2]

Resumo: Este artigo discute técnicas de tolerância à falhas para componentes programáveis, conhecidos por FPGAs (*Field Programmable Gate Arrays*). Essas técnicas baseiam-se em modificações a nível de circuito lógico implementadas em descrição de alto nível, sem modificação na arquitetura do FPGA. O método baseado em descrição de alto nível utiliza redundância tripla de módulos (TMR) e a combinação entre redundância dupla de módulos (DMR) com detecção de erros concorrentes (CED), que pode lidar com falhas na parte lógica combinacional e seqüencial. Os métodos foram validados por experimentos de injeção de falhas emulados em uma placa de prototipação. Os resultados foram analisados em termos de confiabilidade, número de pinos de entrada e saída, área e desempenho.

Abstract: This paper discusses fault-tolerant techniques for programmable devices, the well-know FPGAs (Field Programmable Gate Arrays). These techniques can be based on circuit level modifications, implemented at the

1 Engenharia de Sistemas Digitais, UERGS, Est. Santa Maria 2300 – Guaíba
{fernanda-lima@uergs.edu.br}
2 Instituto de Informática, PPGC, DELET, UFRGS, Caixa Postal 15064
{neuberg, carro, reis@inf.ufrgs.br}

high-level description, without modification in the FPGA architecture. The high-level method is based on Triple Modular Redundancy (TMR) and a combination of Duplication Modular Redundancy (DMR) with Concurrent Error Detection (CED) techniques, which are able to cope with upsets in the combinational and in the sequential logic. The methodology was validated by fault injection experiments in an emulation board. Results have been analyzed in terms of reliability, input and output pin count, area and performance.

1 Introdução

O progressivo avanço no processo de fabricação de circuitos integrados tem aumentando significativamente a sensibilidade dos dispositivos eletrônicos a ruídos externos e internos. Os principais fatores que contribuem para esse aumento na probabilidade de erros são as dimensões dos transistores que diminuem gradualmente com o avanço da tecnologia, assim como a tensão de alimentação. Técnicas de tolerância a falhas estão cada vez mais sendo necessárias para assegurar o correto funcionamento dos circuitos nas novos parâmetros de tecnologia. A confiabilidade, que sempre foi uma parâmetro muito importante no desenvolvimento de circuitos para aplicações espaciais, tornou-se uma variável de projeto igualmente importante em circuitos integrados para aplicações terrestres, juntamente com área, desempenho e potência dissipada. Aplicações consideradas de alto risco, como servidores bancários, servidores de telecomunicação, aviões e outros, estão sofrendo o efeito da radiação e devem ser protegidos com técnicas de tolerância a falhas para garantir confiabilidade.

Circuitos programáveis tais como matrizes programáveis no campo da aplicação, conhecidos como *Field Programmable Gate Arrays (FPGAs),* estão sendo cada vez mais demandados por projetistas de circuitos eletrônicos para aplicações espaciais e terrestres devido a sua alta flexibilidade lógica em alcançar múltiplos requerimentos como alto desempenho, baixo custo no desenvolvimento e rapidez de chegada do produto ao mercado. Em particular, FPGAs programáveis por SRAM (*Static Random Access Memory*) são muito valiosos para missões espaciais ou de difícil acesso pois podem ser reprogramados a distância quantas vezes for necessário muito rapidamente. Conseqüentemente, eles oferecem o benefício adicional de mudanças e melhorias no projeto feitas a distância, correções de erros e ajustes após o lançamento espacial. Por este motivo FPGAs programáveis por SRAM foram escolhidos como o tipo de circuito alvo para a investigação de novas técnicas de tolerância a falhas deste trabalho, mais especificamente a família de FPGA Virtex da empresa Xilinx [1].

Este trabalho analisa em detalhas os efeitos das falhas transientes na arquitetura de um FPGA baseado em SRAM e as principais técnicas de proteção a falhas utilizadas recentemente, como por exemplo a triplicação com votação. A técnica de triplicação da lógica com votação, conhecida como *triple modular redundancy* (TMR), combinada com uma reconfiguração constante da programação (scrubbing) é utilizada no FPGA Virtex para proteger este contra falhas. O TMR é uma técnica adequada a FPGA baseado em SRAM por sua característica de redundância espacial completa, ou seja, da parte física (hardware) na lógica seqüencial e combinacional. O circuito final protegido foi testado utilizando injeção de falhas e em um laboratório com gerador de partículas energizadas. Resultados em termos de confiabilidade, área e desempenho são apresentados neste trabalho.

Esse artigo está organizado como mostra a seguir. A sessão 2 apresenta os principais efeitos do choque de uma partícula energizada em um circuito integrado, especialmente tratando de circuitos programáveis customizados por elementos de memória do tipo SRAM.. A sessão 3 mostra o estado da arte em técnicas de proteção contra falhas transientes em arquiteturas programáveis (FPGAs). Uma nova técnica inovadora de proteção contra SEU em circuitos programáveis é apresentada e discutida em detalhes no capitulo 4.

2 Efeitos das falhas em circuitos programáveis por SRAM

As partículas energizadas (íons) presentes no espaço e os nêutrons presentes na atmosfera terrestre são geradores da falhas em circuitos digitais. Um dos efeitos mais famosos é chamado de erro transiente, erro fraco (soft error) ou falha de evento único (Single Event Upset (SEU)). Essas Falhas ocorrem quando uma partícula energizada incide na superfície do circuito integrado (silício) transferindo uma energia suficiente para provocar uma troca de valor em uma célula de armazenamento (latch ou flip-flop) ou um pulso de corrente no circuito combinacional que pode ser interpretado como um sinal [2].

SEU apresenta um efeito peculiar em FPGAs baseados em SRAM quando uma partícula energizada atinge a lógica combinacional do usuário mapeada na arquitetura programável. Esse fato é devido a lógica do FPGA ser composta por uma estrutura regular de células de memória capaz de implementar a lógica combinacional, chamada de *Look-up Table*, de flip-flops para implementar a lógica seqüencial e de uma matriz de células de memória capaz de configurar as conexões internas, construindo assim o roteamento do circuito. A figura 1 mostra o mapeamento de parte de um pequeno circuito combinacional (somador de 1 bit) para a lógica programável do FPGA. Note que todas as células de memória existentes na lógica programável do FPGA são pontos sensíveis às falhas do tipo SEU.

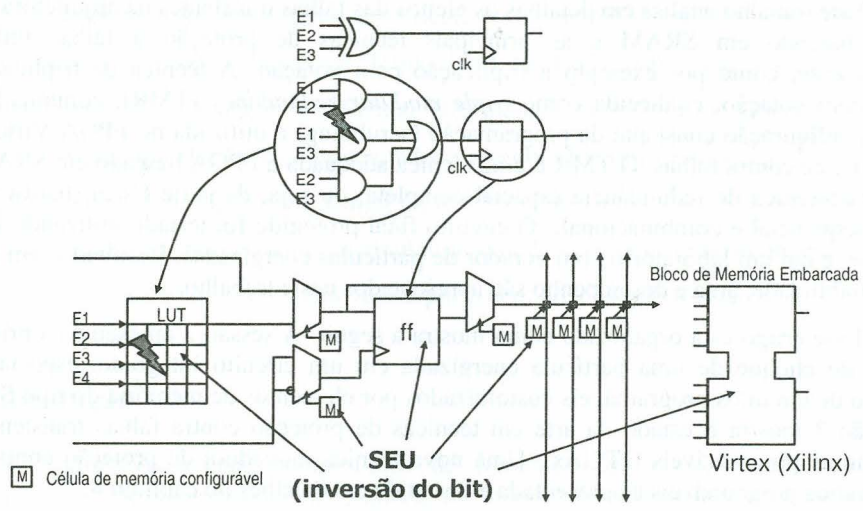


Figura 1. Descrição dos nodos sensíveis à falhas em uma célula programável de FPGA

Em um circuito ASIC, o efeito de uma partícula atingindo a lógica combinacional ou seqüencial é transiente, a única variação é o tempo de duração da falha. A falha no circuito combinacional é um pulso transiente que pode ou não desaparecer de acordo com o atraso na lógica, na topologia e nos vetores de entrada (sensibilização do caminho). Em outras palavras, isso quer dizer que umas falha transiente em uma lógica combinacional pode ou não ser capturada pela célula de armazenamento. Falhas nos circuitos seqüenciais manifestam-se como uma inversão no valor armazenado (bit flip), e irão se manter até a próxima carga da célula de armazenamento [3].

Por outro lado, em FPGA baseado em SRAM, ambas lógica combinacional e lógica seqüencial são implementadas por células de armazenamento (SRAM). Quando uma falha ocorre na lógica combinacional, atingindo o roteamento ou a lógica, ela possui um efeito transiente seguido de permanente porque a célula de armazenamento que compõe aquela lógica ou que controla aquele roteamento teve o seu valor invertido. Esse valor somente será corrigido após a reconfiguração do FPGA. Isso significa que uma falha transiente na lógica combinacional do FPGA tem um efeito permanente e será capturado por uma célula de armazenamento durante a próxima carga, ao menos que alguma técnica de detecção ou correção de falhas seja utilizada. Quando uma falha ocorre na lógica seqüencial do FPGA, o efeito é transiente, igual ao que acontece no ASIC, porque a falha pode ser corrigida na próxima carga da célula de armazenamento. Conseqüentemente, é muito importante levar em consideração o efeito de uma falha transiente (SEU) em FPGA baseado em SRAM no desenvolvimento de técnicas de proteção contra SEU neste tipo de arquitetura.

3 Estado da Arte em técnicas de tolerância a falhas em circuitos programáveis por SRAM

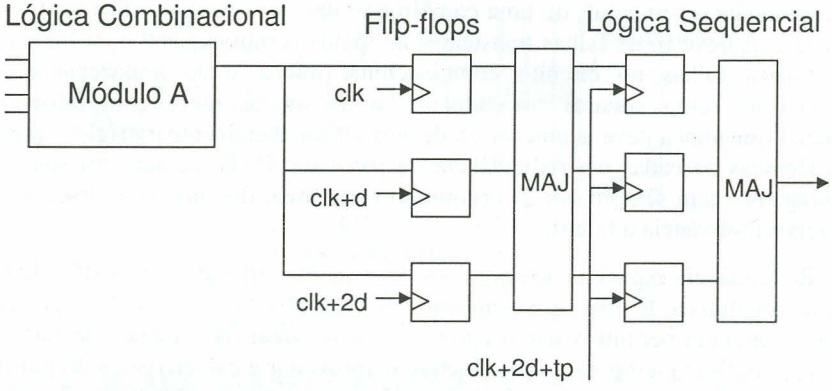
Várias técnicas de proteção contra SEU foram propostas nos últimos anos visando evitar falhas transientes em circuitos integrados. Um circuito imune a SEU deve ser composto por uma variedade de técnicas de proteção baseadas em redundância. Redundância é alcançada através de componentes extra (redundância espacial), de tempo de execução extra (redundância temporal) ou uma combinação das duas. Uma técnica de proteção contra SEU eficiente deve tratar falhas transientes na lógica combinacional e na lógica seqüencial. Desta forma, falhas no circuito combinacional nunca serão armazenados no circuito seqüencial ou serão votados corretamente, o mesmo acontece com falhas no circuito seqüencial que nunca deve acontecer ou devem ser imediatamente corrigidos por votação ou outras técnicas baseadas em redundância ou paridade. Cada técnica tem suas vantagens e desvantagens e tem sempre um compromisso entre área, desempenho, potencia dissipada e eficiência na tolerância a falhas.

Redundância espacial e temporal são largamente utilizadas em ASICs. Elas variam de detecção simultânea de erros a mecanismos de correção. O uso de redundância espacial ou temporal completa permite votar o correto valor do sinal na presença de falha (SEU). No caso da redundância temporal [4], o objetivo é aproveitar a característica do pulso transitório gerado pela falha e comparar o sinal de saída em momentos diferentes. Logo, a saída da lógica combinacional é carregada em três momentos diferentes, onde a transição do relógio da segunda célula de armazenamento é deslocada de um atraso d e a transição do relógio da terceira célula de armazenamento é deslocada de um atraso d vezes 2. Um circuito votador escolhe o valor correto. O esquemático está ilustrado na figura 2a. O aumento em área é devido as células de armazenamento extras e a penalidade em desempenho é devido a captura com atraso máximo de 2 vezes o atraso que é referente ao tempo de duração do pulso. A complexidade deste método é devido aos 3 diferentes relógios.

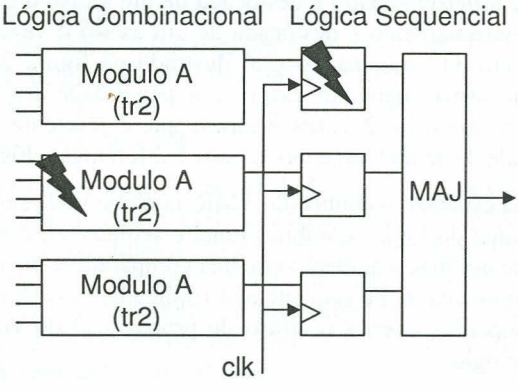
A redundância espacial, o conhecido TMR, também pode ser utilizado para identificar o valor correto na saída da lógica combinacional e seqüencial, como apresentado na figura 2b. Embora apresente um maior aumento em área comparado com a redundância temporal, já que toda a lógica combinacional e seqüencial é triplicada, essa técnica não apresenta grande penalidade no desempenho, apenas o atraso de propagação do votador e não necessita de diferentes fases do relógio.

No caso de FPGA customizado por SRAM, o problema de encontrar uma técnica eficiente de proteção a falhas transientes é ainda mais eminente devido ao grande numero de células de memória SRAM que compõem o circuito (LUTs, bits de programação no CLB e no roteamento, flip-flops e memória embarcada). O objetivo é encontrar o melhor compromisso em termos de área, desempenho, custo e nível de proteção. Há duas maneiras de proteger um FPGA customizado por SRAM: o método arquitetural, onde a topologia da matrix é substituída por uma nova tolerante a falhas, e o método de alto nível, onde a descrição de alto nível de hardware é modificada para ficar tolerante a falhas antes de ser

sintetizada no FPGA. O uso de FPGAs em aplicações espaciais é bem recente e há muito trabalho ainda ser feito. Atualmente, não há uma solução completamente eficiente para FPGAs customizados por SRAM que pode assegurar 100% de confiabilidade com baixo custo em área, alto desempenho e baixo custo de implementação. Este trabalho investigou as técnicas utilizadas atualmente e propôs melhorias para aumentar o grau de confiabilidade e baixar os custos.



(a) Redundância completa no tempo



(b) Redundância completa de hardware

Figure 2 – Exemplos do técnicas de correção de erros transitórios no circuito combinacional e seqüencial.

Neste artigo iremos nos deter apenas nas técnicas implementadas nas descrições de alto nível. As técnicas que modificam a arquitetura do FPGA são muito custosas por necessitar um projeto inicial de todas as partes do FPGA e fabricação de protótipo. Um dos principais motivos de tornar um desafio e desenvolvimento de técnicas tolerantes à radiação

em FPGAs programáveis por SRAM é o fato do efeito dessas falhas ser permanente, como explicado na sessão anterior. Conseqüentemente, soluções de tolerância a falhas usadas em ASICs como códigos de detecção e correção de erros e TMR original com apenas um votador não podem ser utilizadas, porque falhas no codificador ou decodificador ou no votador iriam invalidar a técnica.

Técnicas especiais devem ser desenvolvidas para FPGAs para tratar este efeito. A técnica de proteção contra SEU usada hoje em dia em projetos sintetizados na arquitetura Virtex é basicamente baseada em TMR com reconfiguração continua do bistream (*scrubbing*) no FPGA para evitar acumulo de falhas na matriz. O esquema do TMR usa três circuitos lógicos idênticos (bloco 0, bloco 1 e bloco 2), sintetizados no FPGA e realizando a mesma operação em paralelo com as respectivas saídas sendo comparadas em um circuito votador de maioridade. A técnica TMR é apresentada em detalhes em [5]. As células de armazenamento da aplicação (flip-flops ou latches) são substituídos por três células de armazenamento e multiplexadores implementados por lookup tables (LUT), um para cada. A lógica combinacional assim como os pinos de entrada e saída também são triplicados para evitar qualquer ponto único de falha dentro do FPGA. Desta forma, qualquer falha dentro da matriz pode ser voltada pela estrutura do TMR assegurando o correto valor na saída. A figura 3 ilustra a técnica TMR em um circuito a ser implementado em um FPGA programável por SRAM.

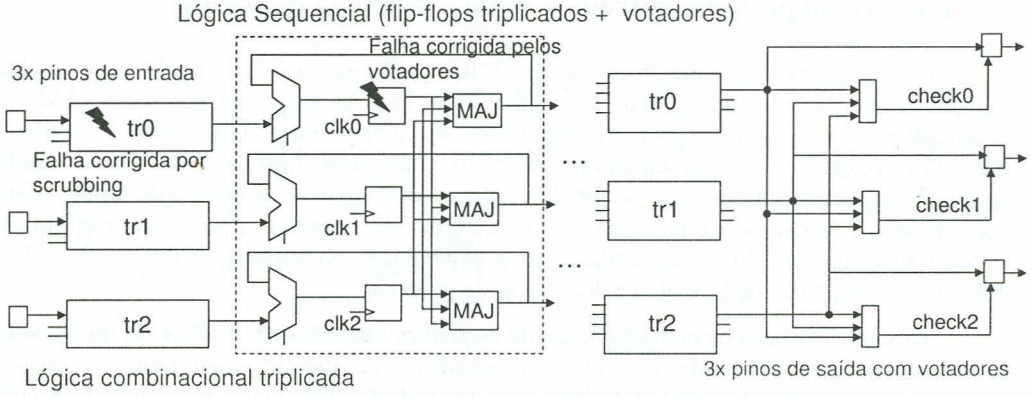


Figure 3. Técnica de TMR implementada numa descrição lógica sintetizada em FPGA.

A técnica TMR foi a primeira a ser testada no FPGA Virtex da Xilinx em um circuito pequeno composto por contadores. Falhas foram injetadas em todos as partes sensíveis da arquitetura e seus efeitos foram detalhadamente analisados. Os resultados de injeção de falha e dos experimentos sob radiação em laboratório comprovaram a eficácia do TMR em proteger circuitos sintetizados em FPGAs customizados por SRAM. Visando testar circuitos mais complexos protegidos por TMR, que incluíssem memória embarcada e um maior numero de lógica, a mesma descrição VHDL do micro-controlador 8051 foi agora protegida por TMR, sintetizada e testada em FPGA. Os mesmos métodos de injeção de falhas e

experimento sob radiação em laboratório foram realizados. Os resultados mostraram que o TMR pode recuperar quase 100% das falhas ocorridas na matriz [6]. Esse numero depende do posicionamento dos blocos redundantes na matriz para evitar que falhas no roteamento afetem mais de um bloco redundante. Embora essa técnica mostre uma alta confiabilidade, ela possui algumas limitações como aumento em área, uso de 3x mais números de pinos de entrada e saída (E/S) disponíveis para a aplicação e conseqüentemente, aumento na dissipação de potencia.

Todavia, como pode-se observar, a técnica de proteção a falhas TMR é custosa em termos de área, logo, foi feito um estudo para o desenvolvimento de uma nova técnica de proteção para FPGAs capaz de reduzir o custo em área, sem diminuir a confiabilidade. Essa tese apresenta uma técnica inovadora de proteção contra SEU em FPGA baseado em SRAM capaz de tratar os problemas previamente descritos: o efeito permanente de uma SEU na arquitetura programável e alto custo em área do TMR. O método combinada duplicação com votação e detecção de erro simultânea baseado em redundância temporal e espacial. Note que a reconfiguração constante da programação (scrubbing) é sempre necessária para evitar o acumulo de falhas na matriz.

4 Otimizando as técnicas de tolerância à falhas em termos de área, número de pinos de entrada e saída e potência dissipada

Com o objetivo de reduzir custos no TMR e melhorar a confiabilidade, uma técnica inovadora em alto nível de tolerância a falhas para FPGAs programáveis por SRAM foi desenvolvida, sem modificações na arquitetura do componente. Essa técnica combina redundância espacial e temporal para reduzir custos e assegurar confiabilidade. Ela é baseada em duplicação com um circuito comparador e bloco de detecção concorrente de falhas. Esta nova técnica proposta neste trabalho foi especificamente projetada para tratar o efeito de falhas transientes em blocos combinacionais e seqüenciais na arquitetura reconfigurável, e reduzir o uso de pinos de E/S, área e dissipação de potencia.

As características de confiabilidade de esquemas baseados em TMR e de técnicas de self-checking foram discutidas em [7]. Os resultados experimentais mostraram que quanto mais complexo o modulo, maior é a diferencia em confiabilidade entre os esquemas de *self-checking* e TMR. Em resumo, a implementação de *self-checking* pode atingir um grau mais elevado de confiabilidade comparado ao TMR, quando a lógica usada para o *self-checking* não for superior a 73% do módulo. A idéia de usar a estrutura de self-checking para técnicas de tolerância à falhas pode ser estendida aos FPGAs através do uso técnicas como duplicação com comparação (DWC) combinada como detecção simultânea de erro (CED). A figura 5 mostra o esquemático geral da técnica, chamada de *hot backup* (DWC-CED). O bloco responsável pela detecção simultânea de erro (CED) é capaz de detectar qual dos dois módulos é falho (presença da falha), e conseqüentemente, existe sempre um valor correto na saída do esquemático por causa do mecanismo capaz de selecionar a reposta correta.

No caso de FPGAs programáveis por SRAM, o bloco CED precisa identificar falhas de efeito permanente nos módulos redundantes. O bloco CED trabalha com o objetivo de achar a propriedade da função implementada pelo modulo para que desta forma seja possível identificar o erro na saída (efeito da falha permanente). Há muitos métodos para detectar falhas permanentes, a maioria das soluções são baseadas em redundância temporal e espacial e elas manifestam a propriedade do bloco lógico em análise.

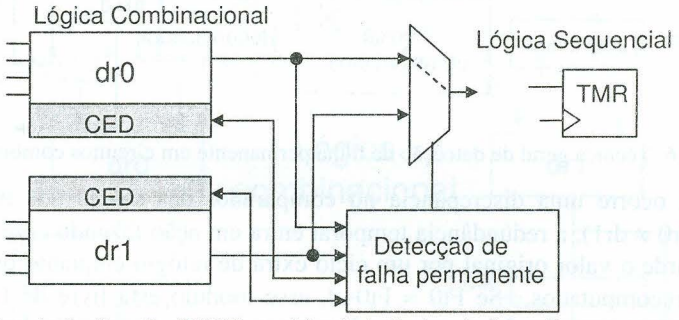


Fig. 5. Técnica de duplicação (DWC) combinada à técnica de detecção concorrente de erro (CED) para a lógica combinacional do usuário representada por dr0 e sua duplicação: dr1.

O esquema CED baseado em redundância temporal recomputa os operandos de entrada de duas maneiras diferentes para detectar falha permanente. Durante o primeiro tempo de computação no tempo t0, os operandos são utilizados diretamente no bloco combinacional e o resultado é armazenado para futura comparação. Durante a segunda computação no tempo t0+d, os operandos são modificados, antes de serem utilizados, de tal maneira que os erros gerados pela falha permanente sejam diferentes aos gerados na primeira computação. Desta forma, a falha pode ser detectada ao comparar os resultados da primeira computação com os resultados da segunda computação. As modificações realizadas nos operandos e nas saídas são chamadas de codificação e decodificação dos operandos e dados e elas dependem das características da lógica do circuito analisado. O esquemático deste método proposto é apresentado na figura 6.

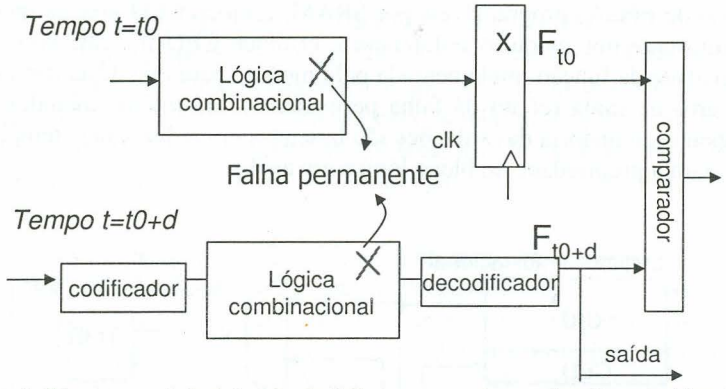


Fig. 6. Técnica geral de detecção de falha permanente em circuitos combinacionais

Quando ocorre uma discrepância no comparador das saídas dos módulos lógicos redundantes (dr0 ≠ dr1), a redundância temporal entra em ação fazendo com que a saída do registrador guarde o valor original por um ciclo extra de relógio enquanto os operandos são codificados e recomputados. Se Ft0 = Ft0+d, esse modulo está livre de falhas. Se Ft0 ≠ Ft0+d), o modulo com falha foi detectado. Uma máquina seqüencial de 4 estados funciona para votar o modulo livre de falha para a saída do esquema. O modulo com falha será corrigido na próxima reconfiguração do FPGA, enquanto isso, a saída final continua recebendo a saída do módulo sem falha.

A combinação da técnica DWC e CED capaz de detectar de falhas permanentes gera uma nova técnica de alto nível para a proteção de falhas transientes em FPGAs. Uma importante característica deste método é que ele não apresenta grandes penalidades no desempenho do circuito porque ele necessita apenas de um ciclo extra de relógio no momento da detecção do bloco sem falha e após continua a operar normalmente. O período do relógio final é igual ao período do relógio original mais o atraso do codificador, decodificador e do comparador de saída. Muitas técnicas de codificação e decodificação foram propostas na literatura para detectar falhas permanentes. [8, 9]. Algumas baseadas em redundância temporal como inversão de bits, recomputação com operandos deslocados (RESO), recomputação com operandos trocados (REWSO), e algumas baseadas em redundância espacial, como predição de paridade e codificação modular.

A figura 7 mostra o esquema do método proposto em um modulo combinacional. Há dois módulos, dr0 e dr1. Existem multiplexadores nas entradas capaz de prover os operandos originais e deslocados. A saída computada na operação normal (operandos originais) é sempre gravada no registrador de amostragem, um para cada modulo. No caso da saída da lógica ser registrada, esse registrador será protegido por TMR, como mostra a figura 8. Nesse caso, cada saída do modulo dr0 e dr1 conecta-se diretamente as entradas dos registradores tr0 e tr1, respectivamente. O registrador tr2 recebe a saída do modulo livre de falha (dr0 ou dr1). Quando inicializado, o circuito inicia passando para o registrador tr2 a saída do modulo dr0. Um comparador na saída do registrador dr0 e dr1 indica uma discrepância (Hc) entre as

saídas de dr0 e dr1. Se Hc=0, nenhum erro aconteceu e o circuito continua funcionando normalmente. No entanto, se Hc=1, um erro é detectado e os operandos precisam ser recomputados para que o modulo livre de falhas seja detectado. A detecção demora um ciclo de relógio como mencionado anteriormente. No caso da saída da lógica não ser registrada, o sinal pode ser enviado diretamente para os pinos de entrada e saída do FPGA (I/O pads) como mostra a figura 9.

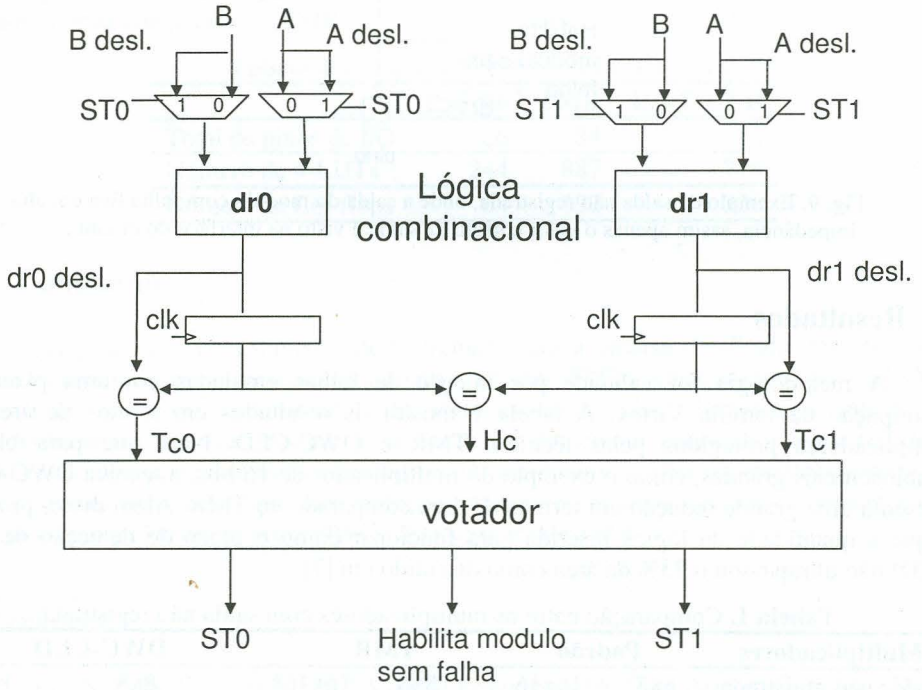


Fig. 7. Redundância no tempo para detecção de falha permanente com circuito de votação baseado no método de detecção concorrente de erro (CED).

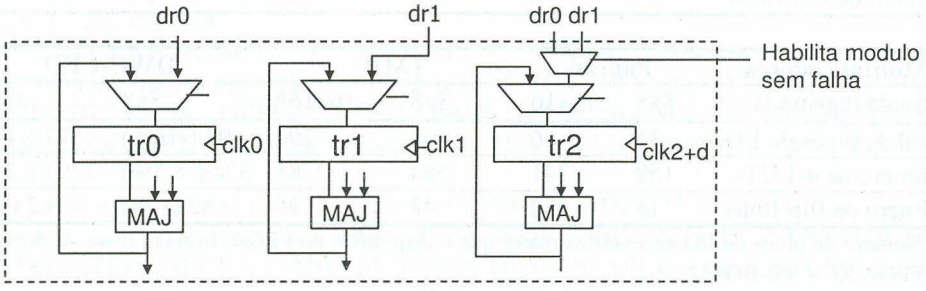


Fig. 8. Exemplo de saída registrada onde a saída é triplicada, um dos módulos recebe a saída de dr0, o outro a saída de dr1 e o ultimo recebe a saída do circuito sem falha.

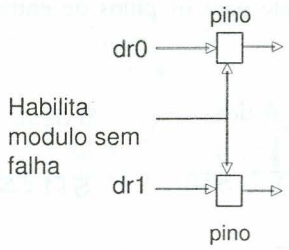


Fig. 9. Exemplo de saída não registrada, onde a saída do modulo com falha fica em alta impedância, assim apenas o valor correto da saída é visto na interface do circuito.

5 Resultados

A metodologia foi validade por injeção de falhas emuladas em uma placa de prototipação da família Virtex. A tabela 1 mostra os resultados em termos de área de multiplicadores protegidos pelas técnicas TMR e DWC-CED. Note que para blocos combinacionais grandes, como o exemplo do multiplicador de 16 bits, a técnica DWC-CED apresenta uma grande redução em termos de área comparado ao TMR. Alem disso, provou-se que a quantidade de lógica inserida para funcionar como o bloco de detecção de erro (CED) não ultrapassou o 73% da área como discutido em [7].

Tabela 1. Comparação entre os multiplicadores com saída não registrada

Multiplicadores	**Padrão**		**TMR**		**DWC-CED**	
Saída não registrada	8x8	16x16	8x8	16x16*	8x8	16x16
Total de pinos de I/O	32	64	96	192	66 (-31%)	130 (-32%)
Número de 4-LUTs	156	711	551	2159	425 (-23%)	1442 (-33%)
Número de flip-flops	0	0	0	0	34	66

Multiplicadores	**Padrão**		**TMR**		**DWC-CED**	
Saída registrada	8x8	16x16	8x8	16x16*	8x8	16x16
Total de pinos de I/O	34	66	108	204	92 (-14%)	172 (-17%)
Número de 4-LUTs	159	741	584	2285	534 (-8,5%)	1791 (-22%)
Número de flip-flops	16	32	48	96	82 (+34)	162 (+66)

* Número de pinos de I/O necessários maior que o disponível no FPGA original, usou-se, portanto, o FPGA XCV300-BG432.

A tabela 2 mostra os resultados em termos de área de um filtro digital na forma canônica protegidos pelas técnicas TMR e DWC-CED. O filtro é de 8 bits e 9 taps, com os coeficientes: 2, 6, 17, 32 e 38. Note que neste caso, há vários multiplicadores e somadores que podem ser protegidos pela técnica proposta. Os registradores continuam sendo protegidos por TMR já que os valores armazenados nos flip-flops devem ser corrigidos pela própria lógica uma vez que o processo de *scrubbing* não atualiza os mesmos. No caso do filtro digital a utilização da técnica DWC-CED também apresentou uma grande redução em termos de área comparado ao TMR.

Tabela 2. Comparação entre os filtros digitais

	Padrão	**TMR**	**DWC-CED**
Total de pinos de I/O	26	84	68
Número de 4-LUTs	244	887	776
Número de flip-flops	64	192	226

6 Conclusão

As principais contribuições deste trabalho são a análise detalhada dos efeitos das falhas transientes na arquitetura da matriz de um FPGA customizado por SRAM, a investigação e teste experimental de técnicas atuais de tolerância a falhas e o desenvolvimento de novas técnicas de proteção que aumentam a confiabilidade e reduzem o custo em comparação com as técnicas atuais. Os resultados de área em multiplicadores e em filtros digitais mostraram que uma redução em área pode ser alcançada ao proteger esses circuitos com a técnica proposta DWC-CED ao invés de TMR.

Referência

[1] XILINX, INC. Virtex®™ 2.5 V Field Programmable Gate Arrays: Datasheet DS003. USA, 2000a.

[2] BAUMANN, R. Soft errors in advanced semiconductor devices-part I: the three radiation sources. IEEE Transactions on Device and Materials Reliability, New York, v.1, n.1, p. 17-22, Mar. 2001.

[3] ALEXANDRESCU, D.; ANGHEL, L.; NICOLAIDIS, M. New methods for evaluating the impact of single event transients in VDSM ICs. In: IEEE INTERNATIONAL SYMPOSIUM ON DEFECT AND FAULT TOLERANCE IN VLSI SYSTEMS WORKSHOP, DFT, 17., 2002. Proceedings... [S.l.]: IEEE Computer Society, 2002. p. 99-107.

[4] ANGHEL, L.; ALEXANDRESCU, D.; NICOLAIDIS, M. Evaluation of a soft error tolerance technique based on time and/or space redundancy. In: SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN, SBCCI, 13., 2000. Proceedings... Los Alamitos : IEEE Computer Society, 2000. p. 237-242.

[5] CARMICHAEL, C. Triple Module Redundancy Design Techniques for Virtex® Series FPGA: Application Notes 197. San Jose, USA: Xilinx, 2000.

[6] LIMA, F.; CARMICHAEL, C.; FABULA, J.; PADOVANI, R.; REIS, R. A fault injection analysis of Virtex FPGA TMR design methodology. In: EUROPEAN CONFERENCE ON RADIATION AND ITS EFFECTS ON COMPONENTS AND SYSTEMS, RADECS, 2001. Proceedings... [S.l.]: IEEE Computer Society, 2001b. p. 275 - 282.

[7] ŁUBASZEWSKI, M.; COURTOIS, B. A reliable fail-safe system. IEEE Transactions on Computers, New York, v.47, n.2, p. 236-241, Feb. 1998.

[8] PATEL, J. H.; FUNG, L. Y. Concurrent Error Detection in ALUs by Recomputing with Shifted Operands. IEEE Transactions on Computer, New York, v.C-31, July 1982.

[9] PATEL, J.; FUNG, L. Multiplier and Divider Arrays with Concurrent Error Detection. In: INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING, 1996. Proceedings... [S.l.]: IEEE Computer Society, 1996.