

## A Method for Design of Impulse Bursts Noise Filters Optimized for FPGA Implementations

Zdenek Vasicek, Lukas Sekanina e Michal Bidlo, Designer, *Automation and Test in Europe - DATE*; Março, 2010;

[www.date-conference.com/proceedings/PAPERS/2010/DATE10/PDFFILES/12.4.4.PDF](http://www.date-conference.com/proceedings/PAPERS/2010/DATE10/PDFFILES/12.4.4.PDF)

Aluna:Maíra Saboia da Silva (098338)

Filtragem de imagem é uma tarefa implementada por muitos sistemas embarcados baseados em FPGA. A qualidade da imagem filtrada influencia todo restante do processamento da imagem. Contudo, filtros com bom desempenho necessitam de mais área no FPGA que filtros padrões. Estudos recentes sobre alguns filtros revelam que algoritmos evolucionários (EA) podem gerar filtros que apresentem a mesma qualidade nos resultados; contudo, com metade do custo no FPGA.

O princípio básico sobre projetos de circuitos evolucionários é que os circuitos eletrônicos são construídos e otimizados por um algoritmo evolucionário com a finalidade de obter uma implementação que satisfaça as especificações dadas pelo projetista. Para obter um circuito candidato, uma configuração em um circuito configurável é criada. Essa configuração é analisada, e um novo circuito é projetado pela aplicação de um operador genético no circuito existente. Circuitos candidatos que tiverem mais parecidos com o projetado, tem maior probabilidade que seu material genético seja selecionado para os próximos circuitos candidatos. O processo de evolução é finalizado quando um circuito ótimo é projetado, ou quando um número máximo de iteração é atingido. Como os EA são processos estocástico, a qualidade do circuito resultante não é garantida no final da evolução.

Esse trabalho aborta um projeto evolucionário de eficientes filtros de área para remover ruídos do tipo *burst* que está comumente presente em imagens de sensoriamento remoto como como imagem de satélite. Ruído *bursts* é um tipo específico de ruído que é difícil de filtrar. Isso é devido por ambos o *pixel* central e seus vizinhos, estarem corrompidos. Então uma janela 5x5 é centrada em cada pixel, e os outros pixels que estão abaixo dessa janela são considerados vizinhos deste. A filtragem proposta, calcula para cada pixel um valor usando o correspondente pixel e alguns de seus vizinhos.

Uma estrutura FIFO (*first in, first out*) foi implementada para ler os valores dos pixels da memória da imagem. A FIFO é tipicamente implementada usando várias BRAMs que servem como *buffer* de linha. Cada BRAMs consegue armazenar 2048 bytes. Ou seja, um *buffer* composto por uma única BRAM é capaz de armazenar imagens com 2048 pixels por linha.

Todos componentes dos filtros são equipados com registradores para suportar filtragem com *pipeline*. O filtro foi implementado em descrição de VHDL usando *Precision Synthesis to Xilinx Vertex II Pro XC2VP50*. O filtro proposto ocupa 128 slices, pode operar a 242 MHz e possui latência igual a 6. Os experimentos foram conduzidos num *cluster* composto por 100 PCs (Pentium IV, 2.4 GHZ, 1GB RAM) usando *Sun Grid Engine (SGE)* que possibilita rodar até 100 experimentos independentes em paralelo.

Após a comparação entre o método proposto e outros métodos presentes na literatura, pode-se constatar que este método apresentou resultados médios de filtragem com alta qualidade nos resultados. Simultaneamente, o custo baixo associado reforça a constatação de que a implementação é favorável; pois, sistemas embarcados de baixo custo representam o alvo dessa aplicação.