

Um sistema reconfigurável para processamento de imagens em tempo real

Luiz Fernando Etrusco Moreira, Julio Cezar David de Melo, José Luiz Silvino

Centro de Pesquisa e Desenvolvimento em Engenharia Elétrica, Universidade Federal de Minas Gerais
{luizf@cpdee.ufmg.br}

Abstract—

This paper proposes the implementation of a reconfigurable system for real-time image processing to analyse the reconfiguration process on high performance computing environment. The system will be implemented using a CMOS image sensor, two reconfigurable devices and an external buffer. One of the reconfigurable devices is responsible for the reconfiguration mechanism while the other is used to implement the image processing dedicated architectures. We show in this paper the hardware design, applications for the system, some performance estimations and system's constraints for using reconfiguration.

Keywords— Reconfigurable Systems, Image Processing, Real-Time Processing

1. INTRODUÇÃO

O processamento de imagens em tempo real envolve inúmeras tarefas que requerem alto poder de processamento. Geralmente, em um ambiente de sistemas embutidos este processamento é realizado através de ASICs nos quais funções previamente escolhidas são incorporadas no chip sem possibilidade de modificações futuras [1]. Com a evolução atual dos dispositivos reprogramáveis, principalmente as FPGAs (*Field Programmable Gate Arrays*), encontramos uma alternativa que fornece bom desempenho, se comparado aos ASICs, somado à flexibilidade para desenvolvimento do sistema [1].

As FPGAs atuais apresentam densidades de até 4.000.000 de portas lógicas permitindo que sistemas complexos sejam implementados [2]. Junto à grande densidade, as FPGAs agregam uma grande flexibilidade tanto na etapa de desenvolvimento quanto na manutenção do sistema já construído, uma vez que podemos reprogramá-las de forma que adaptações, atualizações e novos componentes sejam incorporados ao sistema projetado.

Além de reprogramações estáticas na fase de projeto, as FPGAs permitem que vários sistemas diferentes se utilizem do mesmo hardware, de forma dinâmica. Este processo de reconfiguração durante a execução, ou reconfiguração dinâmica, traz interessantes desafios e novas metodologias de projeto, porém sofre de limitações devido ao tempo gasto para se programar totalmente um dispositivo. Atualmente este tempo é da ordem de 10 ms.

Atualmente diversos fabricantes vêm reduzindo este tempo através de novas tecnologias para que a reconfiguração dinâmica se torne uma metodologia a ser utilizada no desenvolvimento de sistemas embutidos. FPGAs com densidades de 50 mil portas lógicas e tempo de reconfiguração da ordem de 1ms [2].

Outra característica que vem se desenvolvendo nos dispositivos atuais é a possibilidade de se realizar reconfigurações parciais, ou seja, parte do hardware permanece em funcionamento enquanto modificações são realizadas no restante da FPGA. Desta forma o tempo de reconfiguração também é reduzido.

Neste artigo, propomos a implementação de um sistema para processamento de imagens em tempo real utilizando a metodologia de reconfiguração dinâmica, de forma que possamos

avaliar a utilização da mesma em um contexto de computação intensiva, inferindo o impacto do tempo de reconfiguração sobre o desempenho e sobre a funcionalidade do sistema.

Nesta implementação propomos a utilização de um dispositivo genérico que possua tempo de reconfiguração relativamente grande, da ordem de 8 ms, para que possamos estimar em pior caso a possibilidade do uso da reconfiguração.

O restante deste artigo está organizado da seguinte maneira. Na Seção 2 fazemos uma descrição geral do sistema a ser implementado mostrando de forma genérica o objetivo do trabalho. Na Seção 3 descrevemos a implementação do sistema, na Seção 4 fazemos uma introdução aos algoritmos a serem implementados para execução no sistema e na Seção 5 temos as conclusões e indicações de trabalhos futuros a serem implementados.

2. DESCRIÇÃO DO SISTEMA

O sistema proposto será composto basicamente por um sensor de imagem CMOS, uma FPGA, uma EPLD, *buffers* externos e memória FLASH. O diagrama básico do sistema pode ser visto na Figura 1.

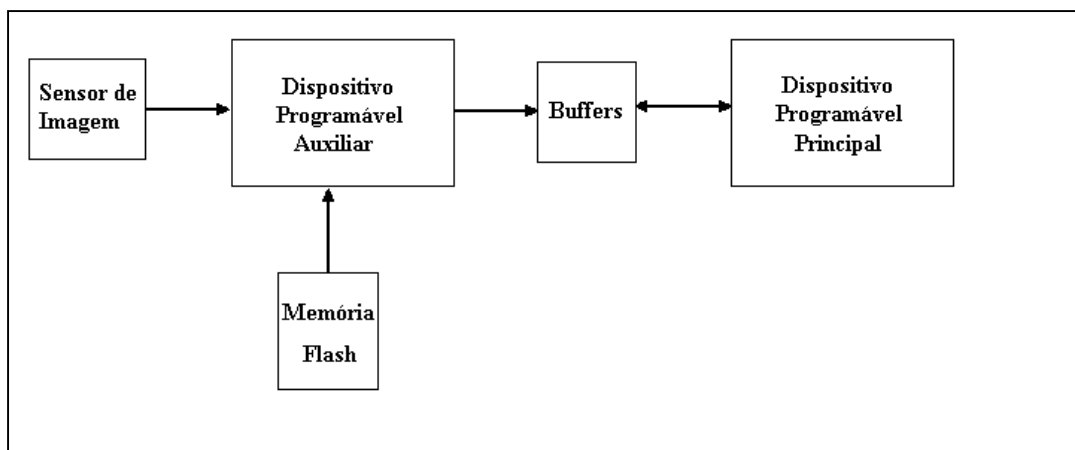


Figura 1: Diagrama Esquemático do Sistema

Nesta configuração deveremos ter imagens sendo constantemente capturadas pelo sensor CMOS, na forma de quadros. Esta captura deverá ser controlada de modo que se possa sincronizar a mesma com a utilização dos *buffers* externos. Porém este controle deve resumir-se apenas em sincronismo, sem interferência no fluxo de dados.

O dispositivo programável auxiliar será uma EPLD e deverá coletar os quadros provenientes do sensor, armazenando-os nos *buffers* externos. Além disso, a EPLD deverá conter um mecanismo de configuração para o dispositivo programável principal de forma que vários algoritmos distintos possam ser carregados da memória FLASH e implementados no dispositivo programável principal. Este mecanismo deverá receber um comando de seleção indicando qual será o algoritmo implementado no momento.

O dispositivo programável principal será uma FPGA na qual os algoritmos de processamento de imagens serão implementados. Deve-se notar que a FPGA deverá acessar os *buffers* e por isso deveremos ter um controlador de *buffers* sempre presente. A maneira para se manter um controlador sempre presente, já que o dispositivo a ser utilizado não deverá possuir características de reconfiguração parcial, será sempre instanciar o controlador de acesso aos *buffers* externos junto aos algoritmos de processamento de imagens. Ou seja, cada algoritmo implementado deverá incluir um controlador de *buffers* para que o mesmo possa ler os dados provenientes do sensor e escrever o resultado do processamento.

Uma operação completa do sistema pode ser descrita através dos seguintes passos:

- 1) os algoritmos a serem executados são armazenados na memória FLASH;
- 2) carrega-se a FPGA com o primeiro algoritmo a ser executado;
- 3) o sensor é inicializado começando a gerar a imagem;
- 4) a EPLD armazena os quadros nos *buffers*;
- 5) a FPGA lê o conteúdo dos *buffers* processando a imagem;
- 6) ao final deste processamento a FPGA tendo armazenado os dados já processados no *buffers*, sinaliza para a carga de outro algoritmo;
- 7) acontece então o carregamento do segundo algoritmo na FPGA;
- 8) a EPLD faz a sincronização de controle de acesso dos *buffers*;
- 9) continua-se nesta seqüência indefinidamente.

Pela operação do sistema descrita acima fica claro a necessidade de utilizar dois *buffers* de entrada em ping-pong e um terceiro para armazenar o resultado temporário do processamento de um algoritmo. Desta forma temos simultaneamente a gravação dos quadros pela EPLD em um *buffer*, a leitura dos dados pela FPGA em outro previamente gravado, e a gravação dos resultados temporários em um terceiro.

O sistema deverá conter ainda uma interface para conexão com um PC, de forma que todo o processo de reconfiguração possa ser monitorado, além de poder-se averiguar os resultados das sucessivas aplicações de diferentes algoritmos sobre as imagens coletadas como, por exemplo, em uma detecção de bordas.

3. IMPLEMENTAÇÃO DO SISTEMA

O sensor de imagem a ser utilizado é um sensor CMOS colorido e possui a resolução de 356x292 pixels, 8 bits por pixel, com taxa máxima de 30 quadros por segundo [4]. Com estas características podemos calcular aproximadamente a taxa de transmissão de dados: $356 \times 292 \times 30 = 2,97$ MBytes/s. Embora esta taxa de transmissão não seja elevada, devido aos requisitos de tempo real do sistema, não podemos interromper a transmissão de quadros, sendo que todo o processamento necessário deve ocorrer concorrentemente à chegada de dados durante a duração de um quadro, ou seja, em 33,33ms.

O tempo de duração do quadro torna-se uma séria restrição haja visto o grande número de operações a serem realizadas em funções de processamento de imagens como, por exemplo, no processo de convolução, uma das aplicações básicas sugeridas na Seção 4.

O sensor entrega através de sua interface uma imagem de 356x292 em padrão Bayer. O padrão Bayer de cores é obtido a partir da colocação de um filtro sobre a matriz de detetores CMOS. Por isto, antes de qualquer processamento devemos fazer primeiramente a conversão do padrão Bayer para a obtenção dos três canais RGB por pixel, acrescida de correção de luminosidade. Em seguida, deve ser realizada a conversão dos três canais RGB para um bitmap de 8 bits, 256 cores e a conversão para 256 níveis de cinza, padrão adequado às aplicações em processamento de imagens pretendidas como, por exemplo, detecção de borda [5][6]. Assim, teremos em uma primeira fase, três algoritmos diferentes a serem executados antes de qualquer processamento específico desejado.

O dispositivo programável auxiliar é a EPLD MAX7128 da Altera constituída de 128 células lógicas [3] e responsável por fazer a interface do sensor com o sistema através do buffer externo. Será também responsável por prover o mecanismo para se realizar a reconfiguração, carregando as aplicações armazenadas na memória flash no dispositivo reprogramável principal. O dispositivo principal é uma FPGA FLEX10K20 da Altera que possui 20 mil portas lógicas [3], na qual serão implementados os algoritmos de imagem

desejados, além dos algoritmos básicos correspondentes às conversões do formato Bayer para canais separados RGB e de RGB para 256 níveis de cinza, 8bits.

Pelos requerimentos de tempo real o *buffer* externo deverá ser capaz de atender à taxa de transmissão de quadros e também deverá possuir um controlador de simples implementação consumindo poucos recursos lógicos, uma vez que o mesmo fará parte de todos aos algoritmos de processamento imagem. Por isso, será implementado com memórias estáticas assíncronas, tanto pela velocidade de acesso quanto pela simplicidade do controlador requerido. O tamanho dos *buffers* deverá ser calculado de maneira que possibilite a operação dos algoritmos de imagem selecionados, e ao mesmo tempo não deverão restringir o tempo de processamento, ou seja, quanto menores os *buffers* maior deverá ser a velocidade de execução dos algoritmos. Inicialmente os *buffers* deverão ser capazes de armazenar um quadro inteiro para posteriormente terem seu tamanho definido. Serão implementados três *buffers* de 128 Kbytes utilizando-se *chips* de memória estática assíncrona na organização de 32K x 8.

O mecanismo de reconfiguração será um simples seletor de aplicações que irá endereçar a memória flash e transferir o conteúdo da mesma para a FPGA. De acordo com as especificações acima chegamos ao sistema que pode ser visualizado na Figura 2.

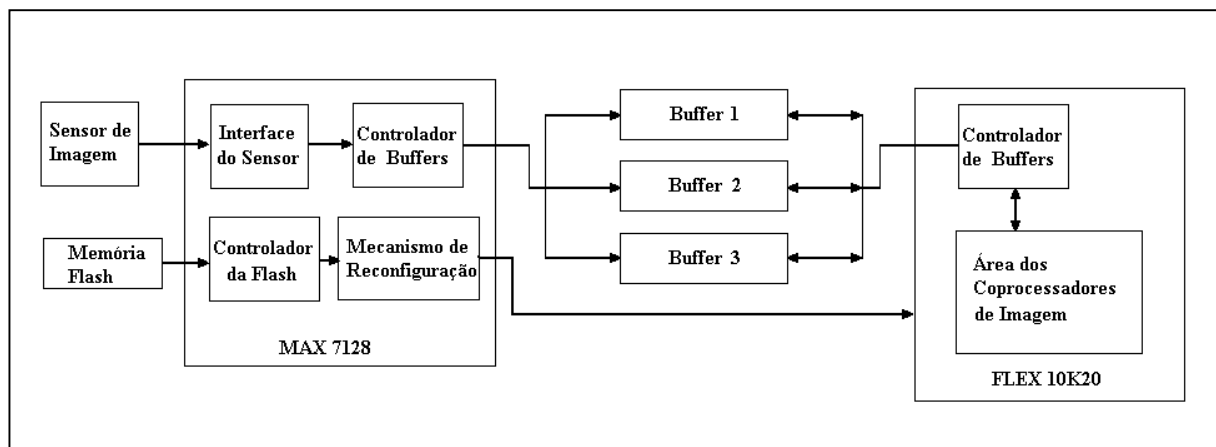


Figura 2: Diagrama Esquemático Detalhado do Sistema

4. APLICAÇÕES

No intuito de analisar o sistema tanto ao nível funcional como em desempenho, selecionamos aplicações em processamento de imagem de maneira a obter uma biblioteca em hardware, para que as mesmas possam ser utilizadas no sistema. No intuito de generalizar a utilização destas aplicações, escolhemos algoritmos básicos de imagem para implementação em hardware. Estes algoritmos são utilizados por várias aplicações diferentes de forma que possamos estender o leque de aplicações a serem executadas.

A aplicação inicial imaginada para o sistema seria a detecção de borda. A detecção de borda é baseada no algoritmo de convolução que é um algoritmo básico em processamento de imagens e pode ser tratado fundamentalmente como um processo de filtragem[5][6]. Neste processo vários filtros podem ser aplicados a uma imagem fazendo a convolução do filtro com a imagem original. A convolução 2D discreta pode ser representada pela seguinte equação:

$$\text{Out}(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \text{image}(m, n) \text{mask}(i - m, j - n)$$

onde m , n representam o tamanho da imagem em pixels. Geralmente o filtro ou máscara é menor do que a imagem original variando em um contexto prático de 3×3 até 17×17 . O uso da convolução com máscaras acima de 5×5 torna difícil a manutenção dos requerimentos de tempo real [5] por ser uma operação que necessita de computação intensiva. Imaginando um imagem de 356×292 pixels realizando a convolução com um máscara de 5×5 em uma taxa de 30 quadros por segundo teremos: 2,97 Mpixel/s, sendo que para cada pixel devemos realizar 25 multiplicações e 24 somas. Deste modo teremos aproximadamente 148 Mops/s. A implementação da arquitetura para convolução deverá servir de base para a configuração de vários filtros diferentes, a serem utilizados através da troca das máscaras. Um destes filtros é o filtro Sobel que funciona como um detetor de bordas.

A escolha desta aplicação deverá não só proporcionar um ambiente de computação intensiva como também uma maneira para futuramente explorar-se a reconfiguração parcial aplicada às máscaras. Desta forma estaremos modificando pequenas porções do hardware e simultaneamente modificando totalmente a funcionalidade do mesmo, numa operação em tempo real.

5. CONCLUSÕES

A implementação do sistema descrito acima deverá permitir que análises funcionais e de desempenho sejam realizadas de forma que se possa qualificar o uso da reconfiguração de maneira abrangente no ambiente de sistemas embutidos de alto desempenho. Além desta análise preliminar poderemos explorar novas características dos dispositivos reconfiguráveis recentes como a reconfiguração parcial, substituindo a FPGA principal, e fazendo com que novas funções possam ser implementadas no sistema além de estender a aplicação do mesmo em outras áreas que envolvam processamento de alto desempenho e em tempo real.

REFERENCIAS

- [1] DEMIGNY, Didier et.al. “*How to Use High Speed Reconfigurable FPGA for Real Time Image Processing?*”, Fifth IEEE International Workshop on Computer Architectures for Machine Perception, 2000.
- [2] XILINX. *The Programmable Logic Data Book*, 2000.
- [3] ALTERA. *Data Book*, 1999.
- [4] ST. VV5404 & VV6404 *Mono and Colour Digital Video CMOS Image Sensors Data Sheet*, 1999.
- [5] MYLER, Harley R., WEEKS, Arthur R. “*The Pocket Handbook of Image Processing Algorithms in C*”, Prentice Hall, Inc., 1993.
- [6] GONZALEZ, Rafael C. e WOODS, Richard E., “*Digital Image Processing*”, Addison-Wesley, 1993.