Uma Metodologia de Projeto e Implementação de Operadores para Processamento Digital de Imagens em Tempo Real usando Field Programmable Gate Arrays (FPGA)

MARCELO ALVES DE BARROS

Universidade Federal da Paraíba
Departamento de Sistemas e Computação
Rua Aprígio Veloso, 882 - 58109-970 - CAMPINA GRANDE - PB
email: barros@dsc.ufpb.br

Abstract. This paper presents an efficient approach to implement low-level image processing algorithms using a reconfigurable hardware technology. The Xilinx Field Programmable Gate Array (FPGA) circuits are used. We describe a synthesis method to generate separable 2D filter architectures and its feasibility on Xilinx FPGAs is evaluated. Time and area constraints are specially considered. An architecture is proposed which allows real time operation of large kernel filters. We present the implementation of a set of basic operators used to build current linear and non-linear (morphological) 2D filters.

1. Introdução

Os algoritmos para Processamento Digital de Imagens em Baixo Nível se caracterizam pela exigência de uma grande quantidade de cálculo, em virtude de manipularem a imagem em seu formato « bruto » (matriz de pontos). Eles são geralmente definidos a partir de operações locais, envolvendo uma vizinhança preestabelecida (janela). Essas operações, normalmente baseadas na convolução bidimensional e/ou em transformações lineares e não-lineares, são usadas para construir filtros bassa-baixas e passa-altas, detetores de contornos, filtros da ordem, operadores morfológicos, etc. Estas tarefas de preprocessamento, em função dos requisitos de velocidade, em particular nas situações de processamento em tempo real, exigem o projeto de arquiteturas paralelas dedicadas, freqüentemente usando um modelo arquitetural SIMD (Single Instruction Multiple Data), e a implementação de processadores elementares sob a forma de circuitos integrados dedicados (ASICs - *Application Specific Integrated Circuits*) [ICASSP (1993)].

O projeto de um operador dedicado, especialmente para tarefas de preprocessamento digital de imagens, deve ser compatível com os requisitos da aplicação (densidade de dados, tempo de resposta aceitável, custo final, etc.), assim como com os requisitos dos processos de desenvolvimento (tempo e metodologia de desenvolvimento, tecnologia de implementação física, etc.) [Auguin (1989)]. Tais requisitos motivam a concepção de ambientes de desenvolvimento que permitam a síntese de arquiteturas dedicadas bem como a modelagem e a simulação comportamental dos operadores projetados. No tocante à implementação física, a evolução contínua dos algoritmos e a diversidade das aplicações têm levado à concepção de ambientes de hardware baseados em arquiteturas reconfiguráveis que possibilitem a implementação de diferentes algoritmos em um mesmo suporte físico. As abordagens normalmente empregadas com este objetivo são baseadas no uso de máquinas compostas de um arranjo de múltiplos processadores e uma rede de interconexões reconfiguráveis, dotadas de estruturas de controle baseadas no conceito de instrução [Dours (1991)], [Maresca (1993)], [Baglieto (1994)], [Gealow (1996)]. Tais soluções, quando direcionadas para o processamento digital de imagens em tempo real em situações típicas de automação industrial (60 imagens/segundo), por exemplo, exigem complexidade de desenvolvimento e custo de implementação relativamente altos.

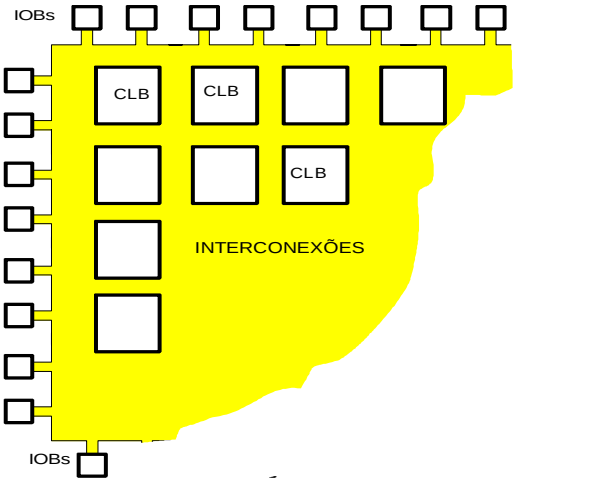
Uma abordagem recente para obtenção de uma melhor relação custo/desempenho consiste na busca de um compromisso entre velocidade e reconfigurabilidade, através de uma adequação entre o algoritmo, a arquitetura e a tecnologia escolhida para a implantação física do operador dedicado [Raimbault (1994)], [Barros (1995)], [Villeumin (1996)]. Neste contexto, a adequação da tecnologia de implementação baseia-se no uso combinado de diversas formas de implementação física (processador de uso geral, DSPs-*Digital Signal Processors*, ASICs e FPGA-*Field Programmable Gate Arrays*) e/ou na exploração da adequação inerente de algumas tecnologias aos algoritmos de processamento de imagens.

Este trabalho baseia-se na abordagem MODARC (*MODular reconfigurable ARChitectures*), apresentada em [Barros (1994)] e [Barros (1995)], a qual considera as características dos algoritmos de processamento digital de imagens em baixo nível e as particularidades envolvendo sua implementação sob a forma de arquiteturas dedicadas, usando circuitos reconfiguráveis FPGA (Field Programmable Gate Arrays). A metodologia aqui apresentada destina-se a sintetizar arquiteturas dedicadas para uma classe importante de algoritmos de filtragem de imagens: os operadores de vizinhança separáveis. Trata-se de uma decomposição da transformada Z dos algoritmos que resulta na estrutura do operador respectivo. Em [David (1988)], esta decomposição foi empregada para a concepção e implementação VLSI de um circuito integrado processador de vizinhança. Neste artigo, apresentamos uma nova abordagem para a implementação física dos algoritmos, otimizada em superfície e em desempenho, bem como uma nova metodologia de projeto baseada no conceito de arquiteturas modulares reconfiguráveis (MODARC), para o caso particular de operadores de baixo nível separáveis. O trabalho está organizado como segue: apresentação da tecnologia FPGA utilizada, descrição da base algorítmica para a síntese dos operadores, apresentação do modelo arquitetural utilizado e da metodologia de projeto e de implementação física, e finalmente, relação de alguns resultados de implementação de operadores elementares representativos.

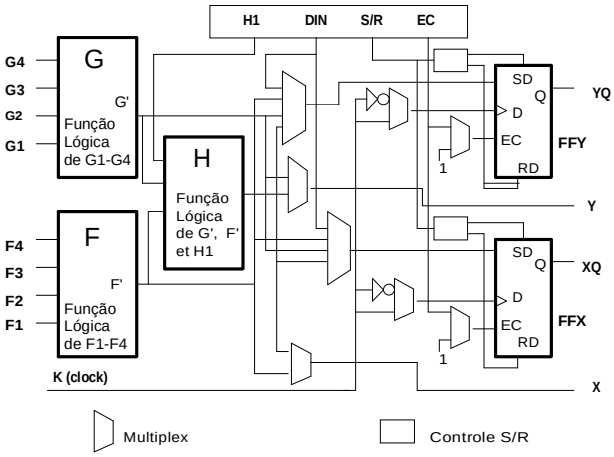
2. Tecnologia de circuitos FPGA SRAM Xilinx

Os circuitos FPGA SRAM são circuitos integrados inicialmente sem funcionalidade, que podem ser configurados pelo projetista e/ou usuário para implementação de uma arquitetura dedicada por ele concebida. Isto é feito pelo carregamento de um arquivo binário (*bit stream*) em um arranjo de células de memória estáticas (SRAM), cujo conteúdo (0 ou 1) define o estado de um elemento reconfigurável do FPGA ou de parte desse elemento. A figura 1.a mostra a estrutura básica de um circuito FPGA Xilinx [Xilinx (1996)]. Trata-se de um arranjo matricial de blocos elementares reconfiguráveis (CLBs) circundado por uma densa rede de interconexões. Esta estrutura é, por sua vez, circundada por blocos reconfiguráveis de entrada/saída (IOBs), associados aos pinos do circuito. A figura 1.b mostra detalhes de um CLB. Um conjunto de ferramentas de projeto de circuitos eletrônicos é disponibilizado pelo fabricante e por construtores associados de software especializados (CAD para eletrônica, simulação, linguagens HDL-Hardware Description Language, etc.) [ViewLogic (1995)]. As principais características desta tecnologia são:

- alto nível de reconfigurabilidade que proporciona ao usuário a possibilidade de implementar a arquitetura dedicada projetada sem a dependência de *founderies* (empresas de fundição de silício);
- redução do tempo e do custo de desenvolvimento de um sistema dedicado;
- necessidade de adequação de suas características às da aplicação para garantir os requisitos de desempenho.



(a)



(b)

Figura 1. Estrutura de um circuito FPGA SRAM da Xilinx e de um bloco elementar reconfigurável do circuito XC4000 (CLB).

3. A metodologia de síntese arquitetural

A imagem *C(m,n)* resultante da convolução bidimensional de uma máscara *LxK* sobre uma imagem *I(m,n)* pode ser expressa pela seguinte equação:

$$C(m,n) = \sum_{i=1}^{L} \sum_{j=1}^{K} I(m-i, n-j).h(i,j)$$

Equação 1

onde:

I(m,n) é a imagem original
h(i,j) é a mascara de convolução de tamanho *LxK*.

Se considerarmos uma imagem em preto e branco, de tamanho *MxN*, representada por níveis de cinza, esta operação demanda *(MxN).(LxK)* multiplicações e *(MxN).(LxK-1)* adições sobre dados codificados em 8 bits. Este número representa uma carga computacional elevada, imposta principalmente pelas multiplicações.

Seja *h(i,j)* um filtro bidimensional cuja função de transferência em Z, pode ser descrita pela equação da figura 2.

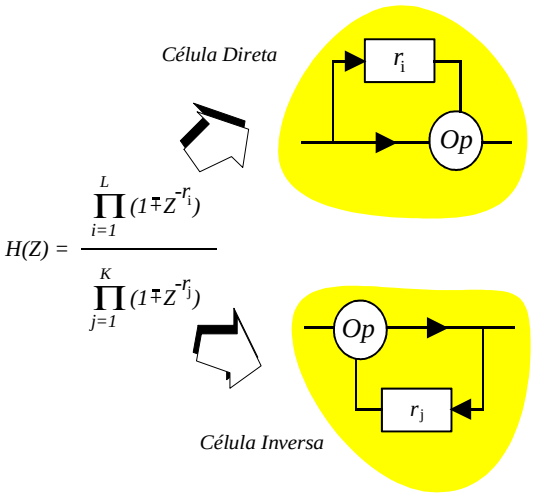


Figura 2. Dedução da estrutura do filtro em função da fatorização de sua função de transferência.

Tal filtro é separável e pode ser implementado a partir de uma estrutura sob a forma de uma cascata de células dos tipos direta e inversa, como mostra a figura 2. Os termos em z representam elementos de atraso de tamanho *r*, os quais podem ser implementados sob a forma de *buffers* ou linhas de atraso, usando registros de deslocamento ou memórias FIFO (*First In First Out*). O termo « separável » indica que a transformada bidimensional pode ser expressa pelo produto de duas transformadas monodimensionais. Se consideramos o exemplo do operador de cálculo do gradiente vertical de Prewitt, temos:

$$\begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix} = \begin{matrix} 1 \\ 1 \\ 1 \end{matrix} * \begin{matrix} 1 \\ 0 \\ -1 \end{matrix}$$

Este filtro é separável e tem a seguinte função de transferência:

$$H(z) = (1+z^{-1}+z^{-2}) \,.\, (1+ 0z^{-1N}-z^{-2N}) \quad ou$$
$$H(z)= (1+z^{-1}+z^{-2}).(1-z^{-2N}) = (1-z^{-1}).(1-z^{-2N}) / (1+z^{-1})$$

Com base na abordagem representada na figura 2, a estrutura para a implementação deste filtro é a mostrada na figura seguinte:

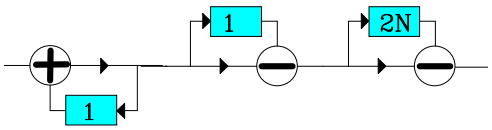


Figura 3. Cascata de células (estrutura) correspondente à implementação do operador detetor de gradiente vertical de Prewitt.

Quando de uma implementação VLSI, esta abordagem leva a uma economia em superfície de silício e a uma redução do tempo de cálculo, uma vez que as multiplicações são substituídas por operações elementares de adição, subtração e/ou lógicas.

Os operadores de morfologia matemática em nível de cinza, tais como dilatação, erosão, abertura, fechamento e suas composições também podem ter suas estruturas deduzidas a partir do mesmo método. Consideremos por exemplo uma operação de erosão monodimensional por um elemento estrutural de tamanho 4, descrita pela equação seguinte:

$$y = min\{x(n) , x(n-1) , x(n-2) , x(n-3)\}$$

Se *Dk* é um elemento de atraso tal que *Dk[x(n)]=x(n-k)*, então temos:

$$y = min\{x(n) , D1\,[x(n)] , D2\,[x(n)] , D3\,[x(n)]\}$$

$$y=min\{min\{x(n),D1[x(n)]\},min\{D2x(n)],D3[x(n)]]\}$$

$$y=min\{min\{x(n),D1[x(n)]\},D2[min\{[x(n)],D1[x(n)]]\}$$

Esta operação pode ser representada pela seguinte estrutura:

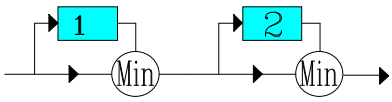


Figura 4. Cascata de células correspondente à estrutura do operador de erosão por um elemento estrutural monodimensional de tamanho 4.

Temos portanto o mesmo padrão de células e de estrutura, apenas com operadores elementares lógicos, não lineares, no lugar de operadores elementares aritméticos. Para filtros morfológicos de tamanho R, a estrutura é constituída de Q células onde Q é o menor inteiro tal que $Q >= log_2 R$.

A condição de separabilidade dos filtros, imposta pelo método de decomposição, não representa uma limitação grave, uma vez que a maioria dos filtros empregados em tarefas de preprocessamento de imagens dispõe dessa propriedade.

4. Otimização do processo de síntese

Com base na propriedade de separabilidade considerada neste método de síntese arquitetural, uma adaptação no algoritmo permite dividir o processamento da imagem em duas etapas. Esta divisão leva a uma otimização do custo de implantação física do filtro. Para ilustrar esta adaptação consideremos um filtro bidimensional com resposta impulsional *h(i,j)*. A convolução de uma imagem *I(m,n)* por *h(i,j)* gera uma nova imagem *C(m,n)* dada pela equação equação 1. Aplicando a transformada em Z a duas dimensões à equação temos:

$$C(z_1,z_2) = \sum_i \sum_j h(i,j) z_1.z_2 I(z_1^i, z_2^j)$$

Se o filtro é separável, a função de transferencia $H(z_1,z_2)$ pode ser escrita como um produto de duas funções de transferência monodimensionais $H_1(z_1)$ $H_2(z_2)$ como segue:

$$C(z_1,z_2) = \sum_i a_i.z_1^i \sum_j b_j.z_2^j I(z_1,z_2)$$

onde a_i e b_j são respectivamente os coeficientes dos filtros monodimensionais $\mathbf{h}_1$ et $\mathbf{h}_2$.

Observando a representação em transformada Z de um filtro separável, podemos notar que H(z) pode ser expressa como:

$$H(z) = H_x(z) \,.\, H_y(z^{-N})$$

O grau de $H_x(z)$ é no máximo *L* e o de $H_y(z^{-N})$ é no máximo *KxN*. Isto implica um custo relativamente elevado da implementação das células que compõem a estrutura do operador H_y, em virtude do tamanho de suas linhas de atraso.

A adaptação do algoritmo baseia-se em uma modificação do modo de acesso aos dados da imagem. Ela consiste em realizar inicialmente uma etapa na direção horizontal (etapa em X), que é implementada pela aplicação da estrutura de cálculo correspondente a $H_x(z)$ segundo uma varredura horizontal (varredura por linha). Em seguida, em uma segunda etapa (etapa em Y), aplica-se a estrutura de cálculo correspondente a $H_y(z)$ sobre a imagem resultante da primeira etapa de processamento, mas segundo uma varredura vertical (varredura por coluna). Esta segunda etapa leva a uma divisão por *N* do grau de $H_y(z)$ e, consequentemente, a uma redução, na mesma proporção (*N* vezes), do tamanho das linhas de atraso das células. Retomando o exemplo do operador Norte-Sul de Prewitt, temos:

$$H_x(z) = (1-z^{-1}) . / (1+z^{-1}) \quad e \quad H_y(z^{-N}) = (1-z^{-2N})$$

Para a aplicação da etapa em Y, através de uma varredura vertical (por colunas), $H_y(z^{-N})$ teria uma expressão adaptada $H_y(z^{-N})^*$, descrita por:

$$H_y(z^{-N})^* = (1-z^{-2})$$

A implementação do filtro corresponderia então à realização de duas cascatas (estruturas) diferentes de células como mostra a figura 5.

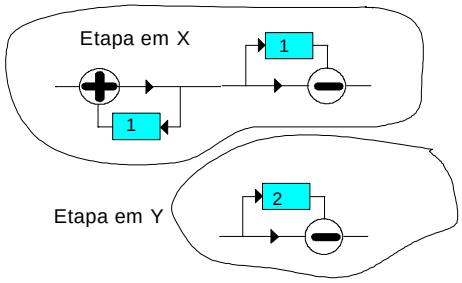


Figura 5. Cascatas correspondentes às etapas em X e em Y para o operador Norte-Sul de Prewitt.

A redução do tamanho das linhas de atraso das células representa uma expressiva economia em superfície em uma implementação VLSI do filtro, especialmente usando circuitos FPGA SRAM, os quais apresentam uma alta relação custo/benefício para a implementação de memória.

5. Arquitetura e metodologia de implementação

O modelo arquitetural para implementação física dos algoritmos separáveis consiste em um *pipeline* de dois processadores realizando respectivamente as etapas em X e em Y (processamentos 1 e 2), e escrevendo (ou lendo) os dados alternadamente em duas memórias distintas de imagem (ver figura 6). Este arranjo possibilita o isolamento dos resultados intermediários e a paralelização dos dois processamentos, gerando uma imagem tratada a cada ciclo de processamento. Além disso ele é adequado ao tipo de fluxo de dados característico de sistemas de visão em tempo real (varredura ou *raster scan*).

A realização desta arquitetura usando o modelo arquitetural de MODARC, ilustrado na figura 7, é trivial, pois o mesmo foi concebido para servir também de suporte à implementação de operadores separáveis de propósito geral. Este modelo consiste de uma cascata de módulos escaláveis de processamento, construídos com um ou vários circuitos FPGA, conectados a bancos de memória (RAM e FIFO), associados aos pares. O *pipeline* de dois estágios proposto para os filtros separáveis pode ser alocado em um par de módulos de MODARC (dois primeiros módulos da figura 7).

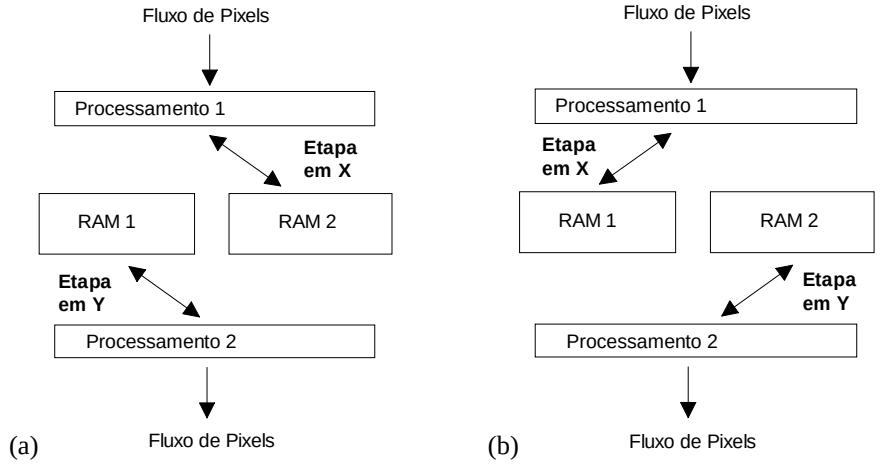


Figura 6. Processamento em duas etapas sob a forma de um pipeline de dois estágios.

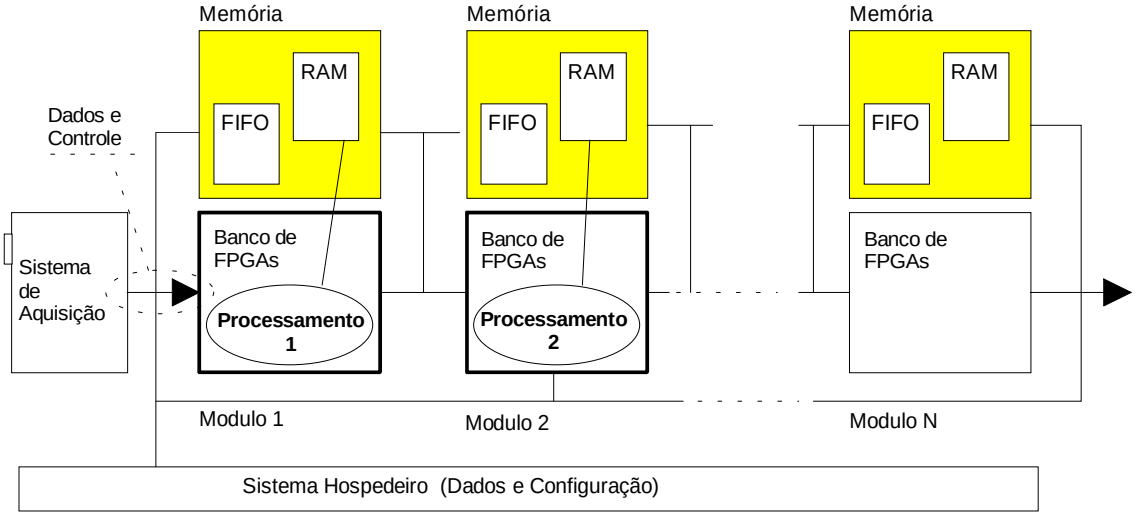


Figura 7. Implementação da estrutura pipeline das cascatas no modelo arquitetural de MODARC

O ambiente MODARC baseia-se na exploração da reconfigurabilidade dinâmica dos circuitos FPGA-SRAM. Nele, um operador dedicado de processamento digital de imagens em tempo real é construído a partir de um conjunto de primitivas elementares de cálculo, de memória e de comunicação. Estas primitivas correspondem a uma biblioteca evolutiva de operadores elementares respectivos, cujo custo e desempenho são conhecidos a priori ou determinados durante o desenvolvimento da aplicação específica. Tais operadores elementares são componentes de hardware dedicados, construídos sob medida, usando porções úteis dos FPGAs dos módulos, bem como os recursos de memória e as estruturas externas de interconexão.

A descrição pode ser feita em diversos níveis (algoritmo, arquitetura e tecnologia de implementação física), usando uma linguagem de alto nível adaptada para a exploração da reconfigurabilidade dinâmica, ou uma linguagem mais adaptada a um método particular de síntese arquitetural empregado.No caso da síntese por decomposição da transformada Z, a descrição de mais alto nível pode ser a simples descrição textual dos coeficientes do filtro e dos requisitos de custo e desempenho da aplicação. O ambiente de síntese se encarrega de gerar automaticamente as duas estruturas do operador dedicado correspondente e o ambiente de implementação gera, em tempo de « compilação », os componentes de hardware respectivos usando os recursos físicos de MODARC. Esta « compilação » corresponde a uma integração e uma automatização de ferramentas de software que produzem um arquivo de configuração da estrutura de interconexão e memórias (reconfigurabilidade arquitetural convencional) e dos FPGAs (reconfigurabilidade dinâmica).

6. Implementação de operadores elementares

No ambiente MODARC, as primitivas usadas na descrição de um operador dedicado correspondem a uma biblioteca de unidades funcionais elementares prefabricadas (precaracterizadas em custo e desempenho) ou fabricadas durante o processo de descrição do operador em desenvolvimento. No caso da síntese por decomposição da transformada Z, uma biblioteca de unidades funcionais elementares (células diretas e inversas) está disponibilizada. A tabela 1 mostra uma relação com o custo e velocidade de células representativas para a construção de operadores separáveis lineares e não-lineares. As células foram implementadas em uma versão protótipo de MODARC que utiliza o circuito FPGA Xilinx XC4010-5, como elemento ativo e memórias LSI-LOGIC de 256Kb, com tempo de acesso de 55 ns.

Nome do operador elementar	Área (CLBs)	Tempo (ns)
ZCELL1+D8	9	18
ZCELL1+D16	13	29
ZCELL1-D8	9	18
ZCELL1-D16	13	29
ZCELL2+D8	13	18
ZCELL2+D16	17	29
ZCELL2-D8	13	18
ZCELL2-D16	17	29
ZCELL1MAX8	13	24
ZCELL2MAX8	17	24
ZCELL1MIN8	13	24
ZCELL2MIN8	17	24

Tabela 1. Custo e desempenho de células dos operadores separáveis implementadas com circuito FPGA Xilinx XC4010-5.

No nome das células implementadas (células ZCELL[x][D/I][y]), D e I indicam o tipo de célula (direta ou inversa), +, -, MAX et MIN indicam o operador da célula, [x] indica o tamanho da célula e [y] indica o número de bits do operador elementar correspondente e de seu barramento (*datapath*). Esta biblioteca, como as bibliotecas de primitivas de MODARC, é evolutiva e pode ser atualizada durante o processo de desenvolvimento de cada nova aplicação.

O operador Norte-Sul de Prewitt (exemplo da figura 5) apresenta um custo de 43 CLBs (uma célula ZCELL1+D16, uma célula ZCELL1-D8 e uma célula ZCELL2-D16) e um tempo de resposta de aproximadamente 30 ns. Apesar de o operador permitir uma freqüência de trabalho de 33 Mhz, sua freqüência máxima de operação é limitada pelo tempo de resposta da memória externa (55 ns) a 18,8 MHz. Devido ao caráter *pipeline* das estruturas geradas, o desempenho do operador é praticamente independente do tamanho do mesmo, dependendo apenas de efeitos de propagação nas interconexões internas do FPGA nos casos de superutilização de sua superfície útil (uso de mais de 75% dos CLBs disponíveis). Este aspecto contribui para a adequação da metodologia apresentada para a implementação de operadores de grandes vizinhanças.

7. Conclusão

A metodologia proposta representa uma maneira eficiente de sintetizar e implementar arquiteturas dedicadas para uma classe importante de algoritmos de

filtragem de imagens. As principais vantagens da abordagem são:

- as operações de multiplicação, normalmente necessárias na implementação dos filtros bidimensionais separáveis são substituídas por operações de adição e subtração, de mais baixo custo de implementação;
- os modelos de cálculo e de paralelismo empregados (cadeia de operadores síncronos em *pipeline*) são adequados ao tipo de fluxo de dados característico do processamento de baixo nível;
- as estruturas geradas são regulares e de fácil implementação VLSI;
- os operadores dedicados são implementados sob a forma de circuitos integrados dedicados, com baixo custo e alto desempenho, permitindo o processamento de seqüências de vídeo em tempo real;
- a metodologia de projeto gerada permite a um usuário de processamento de imagens, sem formação prévia em microeletrônica, projetar e construir um operador dedicado com características de ASIC.

8. Agradecimentos

Este trabalho faz parte de um projeto de desenvolvimento de uma metodologia de Hardware/Software Codesign para sistemas de processamento digital de imagens em tempo real, apoiado pelo programa de fomento à pesquisa do CNPq e pelos projetos *R-Cycle* (*Modelo de Produção, Disponibilização e Evolução de Software*) e *COMATM* (*Comutador ATM para redes de alta velocidade*), do programa Protem-CC.

9. Referências

P. Baglieto, M. Maresca, M. Migliardi. « Pure SIMD processor arrays with a two-dimensional reconfigurable network do not suport virtual parallelism ». Proceedings of the IPPSº 94 Workshop on Reconfigurable Architectures, Cancun, Mexico, April 1994.

M. A. de Barros and M. Akil. « Low Level Image Processing Operators on FPGA: Implementation Examples and Performance Evaluation ». Proceedings of the 12 th International Conference on Pattern Recognition, Jerusalem, Israel, October 1994.

M. A. de Barros. « A High Level Approach to Design and Implementation of Real Time Low-level Image Processing Operators». Proceedings of the IEEE MidWest International Conference on Circuits and Systems, Rio de Janeiro, Brazil, July 1995.

M. Aiguin, F. Boeri, C. Carrieri, G. Menez and A. M. Hugues. "Compilation de machines parallèles VLIM spécialisées", 2ème Symposium ANM, septembre 1989, Toulouse, France.

F. Capello and J. L. Bechennec. "PTAH: Un Réseau de Processeurs à géométrie compilable pour les applications de traitement numérique intensif." III Symposium sur les architectures nouvelles de machines. Juin, 1991, France.

J. C Gealow, F. P. Herrmann, L. T. Hsu and C. G. Sodini. « System Design for Pixel-Parallel Image Processing » . IEEE Trans. on VLSI Systems, Vol. 4, Nº 1, March 1996.

D. David, T. Court, J. L. Jacquot, A. Pirson. "INP20: An image neighborhood processor for large kernels." IAPR Workshop on CV - Special Hardware and Industrial Applications, Tokyo, October, 1988.

D. Dours, R. Facca, A. Feki, P. Magnaud, B. Sautet. "Méthodologie et outil de conception d'une architecture parallèle temps réel". II Symposium sur les architectures nouvelles de machines. Septembre, 1990, France.

M. Maresca, H. Li and P. Baglieto. « Hardware support for fast reconfigurabilty in processor arrays ». Proeedings of the International Conference on Parallel Processing, St. Charles, August 1993.

ICASSP´93 Proceedings of the International Conference on Acoustics, Spech and Signal Processing, 1993.

J. E. Vuillemin, P. Bertin, D. Roncin, M. Shand, H. H. Touati and P. Boucard. « Programmable Active Memories: Reconfigurable systems Come of Age » . IEEE Trans. on VLSI Systems, Vol. 4, Nº 1, March 1996.

F. Raimbault, D. Lavenier, S. Rubini and B. Potier. « Fine grain parallelism on a MIMD machine using FPGAs ». Reserach Repport INRIA Nº1983, May 1993.

Viewlogic User Reference Manual, Viewlogic ProSeries Systems, Inc., Massachusetts, USA, 1995.

The Programmable Gate Array Data Book, Xilinx, San Jose, CA, USA, 1996.