

Arquiteturas Híbridas Reconfiguráveis de Múltiplos Propósitos Baseadas em DSP e FPGA

Éricles R. Sousa¹, José Matias L. Filho¹, Luís G. P. Meloni¹

¹Departamento de Comunicações (DECOM)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6101 – 13.083-852 – Campinas – SP – Brasil

{ericles,matias,meloni}@decom.fee.unicamp.br

Abstract. *This paper aims to describe the use of reconfigurable hybrid architectures for multiple-purpose based on DSP and FPGA. This study makes a comprehensive analysis of efficient methods of code implementation exploiting the computing parallelism of digital signal processing, such as, DCT (Discrete Cosine Transform) and FFT (Fast Fourier Transform) using Distributed Arithmetic. We also show some considerations about an efficient interconnection between DSP (Digital Signal Processor) and FPGA (Field-programmable Gate Array) using shared memory.*

Resumo. *Este artigo tem como objetivo descrever o uso de arquiteturas híbridas reconfiguráveis de múltiplos propósitos baseadas em DSP e FPGA. Este estudo faz uma análise abrangente dos métodos eficientes de implementação de códigos explorando o paralelismo computacional em equações de processamento digital de sinais, como DCT (Discrete Cosine Transform) e FFT (Fast Fourier Transform), baseadas em aritmética distribuída. Também é apresentado algumas considerações sobre a interconexão dos dispositivos DSP (Digital Signal Processor) e FPGA (Fiel-programmable Gate Array) de forma eficiente por meio de memória compartilhada.*

1. Introdução

Nos últimos anos, arquiteturas computacionais híbridas baseadas principalmente na combinação entre os dispositivos FPGAs (*Field Programmable Gate Arrays*) e DSPs (*Digital Signal Processor*), têm evoluído significativamente, alcançando elevados níveis de densidade, altos índices de desempenho e um menor custo de fabricação [LemesFh 2009]. Além dos avanços em capacidade, desempenho e custo, os fabricantes de FPGAs e DSPs têm desenvolvido no decorrer dos últimos anos, ferramentas cada vez mais avançadas de reconfiguração para esses dispositivos.

No caso do dispositivo FPGA existem diversas ferramentas de reconfiguração baseadas no princípio de prototipagem rápida, as quais consistem em *softwares* de desenvolvimento em que os diversos recursos de um sistema podem ser alterados em tempo real, assumindo assim uma característica adaptativa em tempo de execução. A reconfiguração dinâmica permite que as alterações de roteamento dos circuitos da FPGA possam ocorrer durante a operação do sistema com base em algumas características ou fatores determinantes, como por exemplo a oscilação de um *bitstream* de entrada. Em outras

palavras, uma arquitetura reconfigurável pode ser vista como um *hardware* dinâmico, em que a configurações dos seus circuitos implementados em software são carregados e retirados da arquitetura quando necessário, permitindo assim, uma grande flexibilidade de uso do *hardware*. Tais características inerentes às FPGAs, são estendidas a todo um sistema, quando passam a compor uma única plataforma de processamento dedicado, principalmente quando esta é também composta por DSPs, o que faz da arquitetura uma poderosa estrutura flexível e robusta para as mais diversas aplicações.

Arquiteturas reconfiguráveis foram projetadas e vêm sendo pesquisadas ao longo das últimas décadas [LemesFh et al. 2007]. No caso das arquiteturas híbridas, elas são utilizadas para implementar aplicações de diversos projetos e muitas permitem num curto espaço de tempo constituir um complexo sistema em *hardware*, o que faz desse tipo de sistema um modelo inovador, porque atende às necessidades de evolução do mercado.

2. Rotinas Computacionais

A exploração do paralelismo em sistemas embarcados possibilita o aumento do desempenho de rotinas computacionais, as quais devem ser executadas de forma múltipla e em diferentes segmentos, interconectadas de tal modo que possam operar de forma cooperativa [Thirer 2006].

Grande parte das aplicações que envolvem processamento digital de sinais, necessitam de arquiteturas dedicadas, que em muitos casos não permitem reconfigurabilidade. Em nosso estudo descrevemos nas subseções seguintes algumas das principais rotinas computacionais envolvidas em aplicações multimídia, como a DCT que é amplamente utilizada para processamento de imagens, e a FFT que é uma técnica que possibilita a análise da DFT (*Discrete Fourier Transform*) de forma mais rápida envolvendo menos recursos computacionais, ela é muito utilizada para análise de espectro de sinais. Tanto a DCT como a FFT podem ser computadas utilizando técnicas da Aritmética Distribuída (AD), que permite a execução dos cálculos em paralelo de maneira mais eficiente, utilizando menos recursos computacionais.

2.1. *Discrete Cosine Transform* - DCT

Devido a DCT permitir a diminuição das redundâncias estatísticas, ela tem sido um dos principais elementos envolvidos no processo de compressão de imagens ou vídeos digitais, ela transforma um conjunto de pontos do domínio espacial em uma representação no domínio da frequência [Saravanan and Bhaskar 2010].

A DCT é um dos principais componentes envolvidos na compressão de imagens digitais, ela é adotada em vários padrões de esquemas de codificação, como, JPEG, MPEGx e H.26x [Chen et al. 2007]. Ela converte um bloco de *pixels* em uma matriz de coeficientes, decorrelacionando a informação da imagem. Os coeficientes computados são mantidos no início do bloco que contém as informações mais significativas, logo, deve-se garantir que esses coeficientes (de menor índice) sejam armazenados, porque os cossenos fornecidos pela transformada diminuem em amplitude conforme os índices aumentam de valor [Bousselmi et al. 2000].

Para se aplicar a DCT no processamento de imagens é necessário para efeitos de eficiência computacional, dividir a imagem original em matrizes quadradas, como por exemplo, 4x4, 8x8, 16x16, 32x32 ou 64x64. Geralmente se utiliza a divisão em blocos

de 8x8, por fornecerem resultados satisfatórios. A DCT-2D é uma extensão direta da DCT-1D, ela pode ser representada como na Eq. 1 [Saravanan and Bhaskar 2010].

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \cos \left[\frac{\pi(2y+1)v}{2N} \right], \quad (1)$$

onde $[u, v] = \{0, 1, 2, 3, \dots, N-1\}$, $\alpha(u)$ e $\alpha(v)$ são definidos nas equações abaixo.

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & , \text{ para } u = 0; \\ \sqrt{\frac{2}{N}} & , \text{ para } u \neq 0. \end{cases}$$

$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & , \text{ para } v = 0; \\ \sqrt{\frac{2}{N}} & , \text{ para } v \neq 0. \end{cases}$$

Em implementações práticas, quando a DCT é calculada em duas dimensões (DCT-2D), ocorre o consumo de um grande número de multiplicadores, que são rotinas de elevado custo computacional, tornando necessário o uso de arquiteturas computacionais de alto desempenho. Devido as multiplicações demandarem um elevado custo computacional, grande parte dos desenvolvimentos atuais procuram utilizar o menor número possível dessas operações, aumentando o número de somadores [Bousselmi et al. 2000].

Em nossa arquitetura proposta, utilizamos a extensão do conceitos de Unidade Lógica Aritmética (ULA) para paralelizar ao máximo os cálculos utilizando algoritmos que minimizam a complexidade das operações e maximizam a quantidade de dados processados simultaneamente.

2.2. Fast Fourier Transform - FFT

A FFT é um algoritmo complexo que calcula de forma eficiente a transformada discreta de Fourier e a sua inversa. Essa técnica é utilizada em diversas aplicações, como por exemplo, na comunicações de dados, no uso de modulação OFDM (*Orthogonal Frequency Division Multiplexing*), em comunicações sem fio, Wi-Fi (*Wireless Fidelity*) e o WiMAX (*Worldwide Interoperability for Microwave Access*), em comunicações com fio do tipo ADSL (*Asymmetric Digital Subscriber Line*), em processamento digital na compressão de vídeos (H.264) e na resolução de equações diferenciais parciais.

Para aplicações em sistemas embarcados que necessitam o emprego deste algoritmo, juntamente com teorias de convolução e correlação, permitem a implementação de diversas técnicas de filtragens para a eliminação de ruídos ou até mesmo interferência de sinais.

2.3. Aritmética Distribuída - DA

A aritmética distribuída tem sido utilizada como alternativa para o desenvolvimento de muitos cálculos de processamentos digitais. A Eq. 2 representa a sua formulação [White 1989].

$$Z = \sum_{k=0}^K a_k x_k, \quad (2)$$

onde: k refere-se aos índices, $a_k = \{a_0, a_1, a_2, \dots, a_{N-1}\}$ é o vetor dos coeficientes e $x_k = \{x_0, x_1, x_2, \dots, x_{N-1}\}$, são os valores de entrada.

A aritmética distribuída não faz o uso explícito de multiplicadores, portanto, é um caminho eficiente para a computação parcial de um produto interno entre um vetor de dados fixos ou variáveis.

3. Arquitetura Híbrida Baseada em FPGA e DSP

As atuais aplicações de processamento de sinais em sistemas embarcados, necessitam de alta flexibilidade e capacidade de reprogramação. Isso devido as necessidades desejada para que um sistema possa se adequar de acordo com as características de diferentes aplicações. Nesse intuito apresentamos nesse trabalho uma arquitetura computacional para sistemas embarcados composto de DSP e FPGA. Esse tipo de arquitetura tem sido alvo de muitos estudos que exploram o potencial de sistema híbridos.

DSPs são processadores constituídos de uma estrutura dedicada em termos de hardware e software. Esses processadores possuem um conjunto de instruções otimizadas para o processamento numérico de alta velocidade, sendo capazes de representar sistemas complexos em tempo real, podendo executar uma operação de multiplicação e soma ao mesmo tempo em um único ciclo de clock. DSPs permitem grande flexibilidade de programação, possibilitando que uma determinada unidade de processamento seja reutilizada por outro módulo. Por exemplo, um multiplicador utilizado para calcular um filtro FIR (Finite Impulse Response) pode ser reutilizado por outra rotina que calcula uma FFT (*Fast Fourier Transform*) [Tran et al. 2005].

Muitas arquiteturas de DSPs já possuem integradas ao *chip* um Controlador de Acesso Direto à Memória (*DMA Controller*), que permite a transferência de dados sem a interrupção do processador. Essa tarefa se realizada pela FPGA, terá um elevado custo computacional, o que reduzirá a performance do dispositivo [Sousa and Meloni 2010].

FPGAs são extremamente flexíveis, reconfiguráveis e permitem grande capacidade de execução em paralelo de informações digitais. Elas são constituídas de blocos programáveis chamados elementos lógicos e uma hierarquia de interconexões reconfiguráveis que permitem que esses componentes estejam fisicamente conectados. Pode-se configurar esses elementos lógicos para executar funções combinacionais complexas ou simplesmente portas lógicas, como AND [LemesFh 2009]. Atualmente já existem ferramentas de prototipagem rápida que permitem a programação do FPGA em alto nível. Esse mecanismo de programação viabiliza o desenvolvimento, em tempo extremamente mais curto, quando comparado com a programação em linguagem descritivas de hardware. As FPGAs têm sofrido grande evolução, hoje esses *chips* evoluíram ao ponto de poderem ser utilizados na execução de tarefas mais complexas dos processadores digitais de sinais, apesar de que na maioria dos casos ainda é preferível a utilização de DSPs [LemesFh 2009].

3.1. Sistema Proposto

Se opondo à tentativa de afirmar que uma tecnologia é superior a outra, buscamos em nosso estudo fazer uma abordagem de uma arquitetura híbrida capaz de unir o que há de mais vantajoso em ambos dispositivos, DSP e FPGA. Quando há a necessidade de tomada de decisão e processamento em ponto flutuante o DSP programável tende a ser a opção preferida. Entretanto, para o controle de um fluxo de dados a serem processados de forma paralela a FPGA é uma solução muito atrativa. Dessa forma, a nossa proposta de arquitetura para sistemas embarcados ilustrada na Figura 1 é basicamente constituída de uma Unidade de Controle (UC) e uma Unidade de Processamento (UP), contendo respectivamente um DSP e uma FPGA.

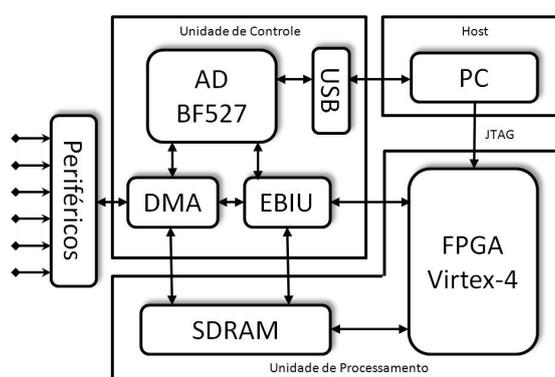


Figura 1. Sistema proposto

A unidade de controle é o núcleo do sistema. A UC é capaz de determinar o fluxo de execução dos dados pela FPGA, controlando a taxa de processamento de forma adaptativa em relação à taxa dos dados de entrada. Sendo ainda, rápida o suficiente para o atendimento das chamadas de interrupções geradas por *hardware* ou *software*. O dispositivo responsável por essa atividade é o DSP Blackfin 527 da Analog Devices. A UC ainda dispõe de um controlador DMA que é usado para transferir dados de forma bidirecional a partir de qualquer origem para qualquer destino dentro da arquitetura, sem utilização do processador para essa atividade.

A unidade de processamento é um módulo constituído basicamente de uma FPGA. Nesse módulo co-existem microblocos de processamento paralelo capazes de executar as operações sob controle do DSP. Para a FPGA, vários registradores de controle e *status* são configurados para que armazenem diferentes informações do processo de comunicação gerenciado pelo DSP. A FPGA que compõe a UP escolhia e a Virtex-4 XC4VLX200 da Xilinx. Esse dispositivo possui alta performance em aplicações lógicas podendo operar em uma frequência de até 500 Mhz.

Foi desenvolvido dentro da FPGA uma máquina de estado responsável pelo gerenciamento de informações vindas da unidade de controle. A etapa da operação de cada microbloco de processamento é ilustrada a seguir:

1. Verifica no registrador de controle é a a próxima da fila de execução (informada pela unidade de controle);
2. Aguarda o sinal para iniciar a operação;

3. Inicia a execução do processo de forma assíncrona aos processos correntes na unidade de controle;
4. Sinaliza para a unidade de controle quais microblocos estão envolvidos na operação e quais estão disponíveis;
5. Após concluído o processamento, cada microbloco sinaliza que está acessível para receber novas orientações da unidade de controle;

De semelhante modo ao descrito em [Sousa and Meloni 2010], a comunicação entre os módulos das unidades de controle e unidade de processamento são realizada usando o conceito de memória compartilhada. O EBIU (*External Bus Interface Unit*) é o controlador já embutido no *chip* do DSP escolhido, que efetua a transferência dos dados mapeados em memória externa baseados em uma lista de prioridades informadas pelo DSP. Esse controlador permite altas taxas de transferência podendo mover blocos de códigos ou de dados para a memória SDRAM que é compartilhada entre o DSP e a FPGA. O EBIU utilizado possui internamente dois controladores de acessos síncronos e assíncronos e pode operar em taxas de frequência de até 133 ciclos de clock.

O Blackfin, baseado em um único cristal, gera dois relógios de controle. O CLK é a referência de tempo para o processador, a partir dessa referência é gerado o SCLK, que é o responsável para gerenciar todos os periféricos. Logo, deve-se considerar que a relação entre esses dois relógios, afetam diretamente o *throughput* dos dados, porque em baixas frequências do CLK, o aumento da latência do barramento do núcleo resultam na redução da utilização do barramento do sistema.

4. Modelos de Aplicações Utilizando DSP e FPGA

Muitas são as aplicações que utilizam arquiteturas baseadas em DSP e FPGA. Diversos estudos descrevem as vantagens de se utilizar esse modelo de estrutura computacional para as mais diversas aplicações.

Em [Dang et al. 2010] é realizada uma análise de arquitetura híbrida aplicada à robótica. Em [Rao and Iyer 2010] o desenvolvimento de um PLL (*Phase-Locked Loop*), que consiste basicamente de um oscilador sintetizador de frequência, apresentou resultados satisfatórios para aplicações *real-time* utilizando um modelo de arquitetura semelhante ao abordado neste artigo.

Outro exemplo de sucesso no uso de arquiteturas híbridas é citado em [MuraliKrishnan and Gangadharan 2010] para compressão de vídeo em alta definição do padrão H.264/AVC, em que a FPGA é utilizada como um co-processador do DSP. No estudo, os autores avaliam e ressaltam os excelentes resultados atingidos com o uso cooperativo entre DSP e FPGA.

Aplicações para IR-UWB (*Impulse Radio Ultra-Wideband*) também podem ser exploradas utilizando a estrutura baseada em DSP e FPGA. Em [George et al. 2009] os autores desenvolveram um *setup* de testes de alta capacidade de processamento. Durante o desenvolvimento do trabalho, foi adotado inicialmente a FPGA como controladora central do sistema, mas a capacidade de implementação de algoritmos no DSP se mostrou muito mais eficiente e a configuração final teve o processador digital de sinais, como o responsável pelo controle de todo sistema, de forma semelhante à recomendada em nosso artigo.

5. Conclusão

Atualmente, podemos constatar em inúmeras fontes bibliográficas, que muitos algoritmos têm sido propostos para a resolução dos mais diversos problemas computacionais. Entretanto, muitos que demonstram vantagens teóricas, muitas vezes possuem a sua implementação prática dificultada, devido à necessidade de grande esforço computacional, o que muitas vezes gera um elevado custo de *hardware*, inviabilizando a implementação de um projeto. O Nosso estudo, apresentou a proposta de uma arquitetura computacional híbrida e reconfigurável para o processamento de diversas aplicações, tornado possível o desenvolvimento de sistemas complexos de forma muito ágil.

Nota-se que muitas aplicações de sistemas embarcados têm utilizado arquiteturas distribuídas, semelhante a que apresentamos neste artigo, devido ao seu grande potencial de reconfigurabilidade e alto domínio da construção e manutenção de códigos complexos, em especial para processamento digital de sinais. Dessa forma, o uso cooperativo entre DSP e FPGA se mostra muito vantajoso, devido a possibilidade de unir em um único sistema as vantagens fornecidas por ambos dispositivos, caracterizando assim, uma arquitetura robusta, eficiente e flexível, dotada de grande capacidade computacional.

Como trabalhos futuros, estamos elaborando alguns testes os quais podem comprovar a eficiência da nossa proposta em cenário distintos, como, por exemplo, na codificação de sinais e também em um cenário compatível com o SBTVD (Sistema Brasileiro de Televisão Digital), onde estamos aplicando e validando o uso de nossa proposta no processamento de rotinas computacionais que visam o cancelamento de eco e retransmissão de sinais, os quais permitem a ampliação do sinal de cobertura por parte das emissoras de televisão digital.

Referências

- Bousselmi, M., Bouhlel, M., Masmoudi, N., and Kamoun, L. (2000). *New parallel architecture of the DCT and its inverse for image compression*. Electronic and Computer Science Laboratory, Tunisia.
- Chen, Y., Cao, X., Xie, Q., and Peng, C. (2007). *An Area Efficient High Performance DCT Distributed Architecture for Video Compression*. The 9th International Conference on Advanced Communication Technology, pp. 238-241, Fev.
- Dang, J., Ni, F., Gu, Y., Jin, M., and Liu, H. (2010). *A Highly Integrated and Flexible Joint Test System Based on DSP/FPGA-FPGA*. International Conference on Robotics and Biomimetics, December, Guilin, China.
- George, G. C., Artiga, X., Moragrega1, A., Ibars, C., and di Renzo, M. (2009). *Flexible FPGA-DSP Solution for an IR-UWB Testbed*. ICUWB 2009, September.
- LemesFh, J. M. (2009). *Estudo de Técnicas de Otimização da Programação de Códigos de DSP em FPGA*. Dissertação de Mestrado, Unicamp.
- LemesFh, J. M., Lenzi, K. G., and Meloni, L. G. P. (2007). *Método Eficiente para Cálculo de uma Função Exponencial em FPGA*. XXVI Simpósio Brasileiro de Telecomunicações, Recife, Pernambuco, Brasil.
- MuraliKrishnan, E. and Gangadharan, E. (2010). *Enhanced Performance of H.264 using FPGA Coprocessors In video surveillance*. 2010 International Conference on Signal Acquisition and Processing.

- Rao, Y. S. and Iyer, S. (2010). *DSP/FPGA Implementation of a Phase Locked Loop for Digital Power Electronics*. IEEE Region 8 SIBIRCON-2010, Irkutsk Listvyanka, Russia, July.
- Saravanan, S. and Bhaskar, V. (2010). *A High Performance Parallel Distributed Arithmetic DCT Architecture for H.264 Video Compression*. 8th International Conference on Electronic Measurement and Instruments.
- Sousa, E. R. and Meloni, L. G. P. (2010). *Exploração de Paralelismo Computacional em Sistemas Embarcados para Compressão de Imagens Digitais*. 1º Simpósio de Processamento de Sinais, Unicamp.
- Thirer, N. (2006). *Parallel Processing for a DSP Application using FPGA*. IEEE 24th Convention of Electrical and Electronics Engineers in Israel.
- Tran, L. H., Mai, T. V., and Molnar, J. A. (2005). *QAM Evaluation System Development: Concept Verification and Hardware/Software Development*. Instrumentation and Measurement Technology Conference.
- White, S. A. (1989). *Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review*. IEEE ASSP Magazine, Vol. 6(3), pp. 4-19, July.