

## AVALIAÇÃO DE APLICAÇÕES DSP PARA USO EM ARQUITETURAS RECONFIGURÁVEIS

**NICOLA, Eduardo V.<sup>1</sup>; RUZICKI, Julio C.M<sup>2</sup>; MATTOS, Júlio C. B.<sup>3</sup>**

<sup>1</sup>Universidade Federal de Pelotas, Curso de Ciência da Computação

<sup>2</sup>Universidade Federal de Pelotas, Programa de Pós-Graduação em Computação

<sup>3</sup>Universidade Federal de Pelotas, Centro de Desenvolvimento Tecnológico  
{evnicola,jcmruzicki,julius}@inf.ufpel.edu.br

### 1 INTRODUÇÃO

Atualmente, o mercado de dispositivos eletrônicos continua em um crescimento acelerado. Os mais diversos equipamentos possuem sistemas computacionais, isto é, possuem normalmente um microprocessador que executa tarefas específicas. O avanço das últimas décadas permitiu o avanço da tecnologia destes processadores juntamente com a demanda por evolução de tais sistemas. Para conseguir tal avanço, novas formas de realizar as aplicações tiveram de ser desenvolvidas, uma delas é o uso de técnicas matemáticas no tratamento de sinais digitalizados.

A técnica chamada de Processamento Digital de Sinais (DSP) realiza um tratamento matemático complexo através de um processador que precisa realizar milhões de operações matemáticas por segundo para ter sucesso em tais aplicações. Os processadores de uso geral, os GPPs (*General Purpose Processors*), não possuem uma arquitetura especializada para tal fim, portanto, processadores especializados foram desenvolvidos para este fim. Estes processadores são chamados de processadores para Processamento Digital de Sinais ou simplesmente processadores DSPs.

Os processadores DSP, possuem suas arquiteturas otimizadas. Dentro destes, existem unidades funcionais capazes de acelerar a execução de cálculos necessários para o processamento digital de sinais (LAPSLEY, 1997), tais como: unidades de ponto flutuante, instruções MACs (*Multiply and Accumulate*), memórias circulares, FFT (*Fast Fourier Transform*), FIR (*Finite Impulse Response*), entre outras. Com tal arquitetura tão especializada, estes processadores ganharam o mercado e estão presentes nas mais variadas aplicações do cotidiano das pessoas, como: multimídia, telefonia celular, instrumentação, eletrodomésticos, equipamentos médicos, etc.

Além destes processadores, para realizar tais aplicações também podemos utilizar: processadores de uso geral (GPPs) ou dispositivos projetados e customizados para tal fim os ASICs (*Application Specific Integrated Circuits*). Os GPPs foram os dispositivos utilizados antes da idealização dos processadores DSPs, sua vantagem está no custo reduzido e sua grande desvantagem está no baixo desempenho quando em aplicações DSP. Já os ASICs possuem custo unitário ainda menor que os GPPs, porém o custo do projeto é grande mas seu desempenho é alto. Os processadores DSP apresentam o melhor custo benefício (intermediário aos GPPs e ASICs): baixo custo unitário, baixo consumo de energia e alto desempenho em aplicações DSP (LAPSLEY, 1997; DINIZ, 2010).

Mesmo os processadores DSP sendo tão bons, ainda existe a necessidade da evolução destes dispositivos e a indústria não consegue mais garantir tal evolução na mesma velocidade e capacidade de processamento

previstos pela Lei de Moore. Tal limitação é dada pela dissipação de calor, que se encontra no limite, gerada pelo aumento da frequência dos processadores (FLYNN e HUNG, 2005). Portanto, para garantir a evolução são necessárias novas técnicas e metodologias são necessárias.

Uma das áreas que vem crescendo em pesquisa é a de Arquiteturas Reconfiguráveis. Esta área de pesquisa pretende aumentar o desempenho dos processadores sem aumentar a frequência de operação dos mesmos através do uso de estruturas feitas em hardware para realizar porções de código do programa os quais demandam muito tempo para serem realizados pelo método computacional convencional (execução de uma sequência de instruções), juntando a personalização propiciada por um ASIC com o custo de um GPP.

Este trabalho encontra-se no contexto de um trabalho maior onde pretende-se demonstrar que através do uso da técnica de reconfiguração incorporada a um processador DSP pode-se acelerar a execução de algoritmos de processamento digitais de sinais. Contudo este artigo possui como objetivo realizar a seleção de aplicações características de DSP, selecionar uma ferramenta que permita o desenvolvimento, a simulação de aplicações e extração de resultados de desempenho para diferentes processadores (diferentes características) para posterior uso na arquitetura reconfigurável.

## 2 METODOLOGIA (MATERIAL E MÉTODOS)

O trabalho seguiu a seguinte sequência de passos: seleção de aplicações características de DSP, seleção uma ferramenta que permita o desenvolvimento, a simulação de aplicações e extração de resultados de desempenho para diferentes processadores, simulação/execução das diferentes aplicações e por fim a análise dos resultados.

Foram avaliadas três ferramentas comerciais de diferentes fabricantes: Code Composer, Symphony Studio e VisualDSP++. Code Composer é um ambiente de desenvolvimento voltado para a família de processadores embarcados da Texas Instruments que compreende ferramentas de implementação, depuração, medida de desempenho (*profiler*), sistema de operação em tempo real, entre outras. A Symphony Studio da Freescale, prove suporte à família de processadores Freescale e é um ambiente executado em conjunto com o software Eclipse e se beneficia de alguns recursos como depuração GDB (*GNU Project Debugger*) para os processadores DSP563xx/DSP567xx e gerenciador de projetos. A ferramenta VisualDSP++ da Analog Devices possui inúmeros recursos, dentre eles, compilador C/C++, medida de desempenho estatística (*statistical profiling*), simulador, montador, ligador e suporte a emulação.

Após a instalação das ferramentas foi realizada uma avaliação de cada ferramenta com os seguintes critérios: possibilidade de compilação de aplicações, simulação de aplicações e resultados fornecidos. Por fim, foi decidido optar pela ferramenta VisualDSP++ (Analog, 2012a) da Analog Devices pois esta dispõe de uma variedade de famílias e modelos de processadores DSP, além de possuir os suporte às simulações pretendidas. Já as ferramentas Code Composer e Symphony Studio possuem serias restrições em relação a necessidade de licenças (que devem ser compradas) para uso de diversas funcionalidades.

Para a seleção das aplicações com características DSP se optou por utilizar aplicações exemplo já disponíveis na ferramenta selecionada. Em um

primeiro momento foi selecionado um conjunto de aplicações mais simples apenas para ambientação com a ferramenta e verificação se a ferramenta forneceria os resultados desejados. A Tabela 1 apresenta as aplicações selecionadas.

**Tabela 1 – Aplicações Selecionadas**

Num	Aplicação
1	Multiplicação de dois vetores reais
2	Adiciona duas matrizes bidimensionais
3	Subtrai duas matrizes bidimensionais
4	Multiplica os elementos de uma matriz por um escalar
5	Calcula a média de uma matriz
6	Calcula o valor RMS de uma matriz de dados
7	Acumula o produto de duas matrizes
8	Adiciona duas matrizes reais

Após a execução das aplicações apresentadas na Tabela 1 serão executadas as seguintes aplicações já selecionadas que representam aplicações DSP de maior porte: filtro biquadrático, filtro FIR e IIR, DCT (Transformada Discreta do Cosseno).

Para execução/simulação das aplicações foram selecionados dois processadores da Analog Devices, um da família Blackfin (ADSP-BF536) e outro da família Shark (ADSP-21477). A família Blackfin (Analog, 2012b) de processadores foi desenvolvida para aplicações de baixo custo mas que exigem altas taxas de processamento digital de sinais, tais como: vídeo, instrumentação industrial, biometria e controle industrial. Dentre muitas de suas características podemos ressaltar as seguintes: frequência de relógio de até 400MHz, instrução MAC de 16 bits (possibilidade de executar até 800 milhões de MACs por segundo), duas ULAs (Unidade Lógica e Aritmética) de 40 bits, quatro ULAs especializadas para vídeo de 8 bits, deslocador de 40 bits. Possui uma arquitetura RISC com um pipeline de 10 estágios.

Os processadores SHARK (Analog, 2012c), de 32 bits, são baseados na arquitetura super-harvard que estende os conceitos originais de barramentos separados para memória de programa e de dados, adicionando um processador de entrada e saída para seus barramentos dedicados. Além destas, podemos citar como características comuns de sua arquitetura: unidade aritmética de 32/40 bits de ponto flutuante; multiplicadores de 32 bits, de ponto fixo; ausência de pipeline aritmético, todas as computações são executadas em um ciclo; endereçamento de buffer circular suportado em hardware; 32 buffers circulares com ponteiros de 32 bits; conjunto de instruções suporta: aritmética condicional, manipulação de bits, divisão e raiz quadrada.

### 3 RESULTADOS E DISCUSSÃO

Os resultados foram obtidos através da simulação das aplicações selecionadas (Tabela 1) na ferramenta VisualDSP++ com os processadores das duas famílias (Blackfin e Shark). Os algoritmos simulados encontram-se com os seus originais na linguagem C. Os resultados de desempenho (apresentados em número de ciclos) para o processador ADSP-BF536 da família Blackfin são em

ordem de grandeza de pior desempenho em relação ao processador ADSP-21477 da família Shark utilizando SISD (*Single Instruction, Single Data*) são apresentados na Tabela 2. Apesar de uma grande diferença isto é esperado devido a família Backfin ser uma família bem mais simples que a família Shark.

Também foi realizada a comparação de execução dentro da família Shark entre os processadores ADSP-21477 (utilizando SISD) e ADSP-21477 (utilizando SIMD – *Single Instruction, Multiple Data*). Os resultados apresentaram melhor desempenho, porém com uma execução de apenas 50%. Com uso de SIMD normalmente espera-se resultados melhores, porém isto é muito dependente das aplicações.

**Tabela 2** – Desempenho das aplicações em diferentes processadores

Aplicação	ADSP-BF536 SISD (ciclos)	ADSP-21477 SISD (ciclos)	ADSP-21477 SIMD (ciclos)
1	69371	2008	1012
2	47171	1808	912
3	48071	1808	912
4	59772	908	466
5	47460	1013	517
6	89483	2039	1044
7	91371	1009	518
8	52371	2008	1012

## 4 CONCLUSÃO

Este trabalho apresentou a seleção de um conjunto de aplicações com características de DSP, de uma ferramenta para simulação de aplicações em diferentes processadores e os resultados de execução para dois diferentes processadores. Pretende-se ampliar o conjunto de aplicações para posterior uso da técnica de reconfiguração incorporada a um processador DSP.

## 5 REFERÊNCIAS

LASLEY, P. **DSP Processor Fundamentals-Architectures and Features**. New York: IEEE Press, 1997.

DINIZ, P. S. R., da SILVA, E. A. B., NETTO, S.L. **Digital Signal Processing: System Analysis and Design**. Cambridge: Cambridge University Press, 2010.

FLYNN, M. J., HUNG, P. Microprocessor design issues: thoughts on the road ahead. **IEEE MICRO**, v. 25, n. 3, CA, USA, p.16-31, 2005.

Analog Devices. **VisualDSP++ Development Software**. Disponível em: <<http://www.analog.com/en/processors-dsp/blackfin/vdsp-bf-sh-ts/products/product.html>>. Acesso em: 22 Jul 2012.

Analog Devices. **Blackfin ADSP-BF536: Data Sheet**. Disponível em: <[http://www.analog.com/static/imported-files/data\\_sheets/ADSPBF534\\_BF536\\_BF537.pdf](http://www.analog.com/static/imported-files/data_sheets/ADSPBF534_BF536_BF537.pdf)>. Acesso em: 22 Jul 2012.

Analog Devices. Shark **ADSP-21477: Data Sheet**. Disponível em: <[http://www.analog.com/static/imported-files/data\\_sheets/ADSP21477\\_21478\\_21479.pdf](http://www.analog.com/static/imported-files/data_sheets/ADSP21477_21478_21479.pdf)>. Acesso em: 22 Jul 2012.