

IMPLEMENTAÇÃO EM FPGA DE TRANSFORMADA WAVELET PARA COMPACTAÇÃO DE SINAIS ELÉTRICOS DE SISTEMAS DE POTÊNCIA UTILIZANDO PROCESSADOR EMBARCADO

EDER BARBOZA KAPISCH*, LEANDRO RODRIGUES MANSO SILVA*,
CARLOS HENRIQUE NASCIMENTO MARTINS*, LUCIANO MANHÃES ANDRADE FILHO*,
CARLOS AUGUSTO DUQUE*

**Universidade Federal de Juiz de Fora
Juiz de Fora, Minas Gerais, Brasil*

Emails: eder.kapisch@engenharia.ufjf.br, leandro.manso@engenharia.ufjf.br,
chnmartins@gmail.com, luciano.ma.filho@gmail.com, carlos.duque@ufjf.edu.br

Abstract— This work aims to present an implementation of the Wavelet Transform on FPGA (Field Programmable Gate Array). Through this tool, a fast compression of electrical signals is possible, smoothing or enhancing the details thereof. Furthermore, it compares this tool implemented in an embedded processor with a direct implementation that uses various logical blocks, called parallel method. This comparison is performed considering the number of logical blocks and the speed for each case. A prototype was developed to deploy both methods and it was able to accomplish this compression, in which more than 85 % of the samples acquired at a rate of 7.68 kHz could be replaced by zero, maintaining over 99.9 % of the original signal energy. The error in the reconstruction was limited to the range of $10^{-2}\%$.

Keywords— Wavelet Transform, embedded processor, FPGA, compression.

Resumo— Este trabalho tem como objetivo apresentar uma implementação da Transformada Wavelet em FPGA (*Field Programmable Gate Array*). Através dessa ferramenta, é possível a compactação rápida de sinais elétricos, suavizando ou realçando os detalhes dos mesmos. Além disso, compara-se essa ferramenta embarcada em processador com uma implementação direta, denominada de método paralelo, cujos diversos blocos lógicos compõem a estrutura não-processada. Essa comparação é feita considerando-se a quantidade de blocos lógicos e velocidade despendida em cada caso. Um protótipo desenvolvido foi capaz de implementar os dois métodos e de realizar esta tarefa de compressão, na qual mais de 85% das amostras adquiridas a uma taxa de 7,68 kHz puderam ser substituídas por zero, mantendo-se mais de 99,9% da energia do sinal original. O erro na reconstrução limitou-se à ordem de $10^{-2}\%$.

Palavras-chave— Transformada Wavelet, processador embarcado, FPGA, compactação.

1 Introdução

Atualmente existe uma crescente necessidade de caracterizar corretamente o comportamento de sistemas de energia elétrica. Cerca de duas décadas atrás, devido à desregulamentarização do setor elétrico em todo o mundo, usuários e empresas públicas passaram a se preocupar com os impactos causados por problemas de qualidade de energia. Esses problemas surgem como resposta ao uso massivo de cargas não-lineares e equipamentos eletrônicos em residências, centros comerciais e instalações industriais. Além disso, foi percebido que essas deficiências poderiam se intensificar caso não fossem corretamente corrigidas. Várias tecnologias têm surgido com o intuito de monitorar o comportamento de sistemas de energia elétrica em diferentes níveis: alta tensão (HV), média tensão (MT) e baixa tensão (LV) (Tcheou et al., n.d.).

Além do monitoramento dos parâmetro elétricos dos sistemas de energia, uma outra aplicação que também envolve processamento de sinais é a compressão dos dados provenientes dos sinais elétricos medidos, cujo objetivo é proporcionar uma transmissão ou armazenamento de dados de forma mais eficiente, visando um

baixo custo computacional. A compressão consiste na eliminação de redundâncias do sinal através do uso de alguma transformada, geralmente transformadas integrais.

A *wavelet* é um sinal base com determinadas propriedades como, por exemplo, energia finita, o que a torna adequada para decomposição de outras funções (DA SILVA, 2008). Seu princípio reside na decomposição hierárquica de um sinal de entrada para uma série de sucessivos sinais de resolução mais baixa. Isso é eficaz pois, ao ser aplicada em um sinal, evidencia seus detalhes, possibilitando a visualização de distúrbios variantes no tempo. Para intervalos de tempo em que a função muda rapidamente, o método pode ampliar a área de interesse para melhor detecção das características do sinal, tornando possível, por exemplo, a análise de transitórios e harmônicos, (Wagner et al., 1993) e (Farge et al., 1996).

Durante muito tempo, as *wavelets* de Haar foram a única base ortonormal de *wavelets* conhecida. Stephane Mallat, em 1985, impulsionou o surgimento de novas funções, com o mesmo propósito, através de seu trabalho em processamento digital de imagens. Pouco tempo depois, Ingrid Daubechies construiu

um conjunto de bases ortonormais de *wavelets* suaves, com suportes compactos e esse trabalho é considerado um dos alicerces das aplicações atuais (de Lima, 2002).

A Transformada Wavelet tem sido aplicada nas mais diversas áreas, por meio de algoritmos rápidos, fundamentais, tanto para compressão de sinais, como para a remoção de ruído. Mesmo sendo uma técnica relativamente nova, tem atraído a atenção de muitos estudiosos, pois se trata de uma ferramenta muito versátil e de conteúdo matemático muito rico. Principalmente, se pudermos escolher as *wavelets* que melhor se adaptem aos dados, ou truncarmos os coeficientes da *wavelet* menores do que um valor previamente estabelecido, ou seja, um limiar, os dados podem ser esparsamente representados. Essa característica faz dessas funções uma excelente ferramenta no campo de compressão de dados (Huang et al., 2002).

Uma forma de implementação ou realização dessa ferramenta pode ser obtida através da FPGA, nesse processo é possível definir vários blocos de *hardware*, que operam em paralelo, acarretando, dessa forma, um aumento significativo na capacidade computacional do sistema (Huang et al., 2002).

Desde a sua invenção em 1984 pela Xilinx®, as FPGAs passaram de um simples chip de silício de lógica personalizável até verdadeiros substitutos de circuitos integrados em aplicações específicas (*application specific integrated circuits - ASIC*) para processamento de sinais e aplicações de controle.

A tecnologia presente na FPGA se destaca por oferecer alguns benefícios únicos, tais como a capacidade de reconfiguração e alta densidade de elementos lógicos. Os circuitos lógicos são desenvolvidos por meio de linguagens descritivas de hardware, dentre as quais podem ser citadas a VHDL (*Very high speed integrated circuit (VHSIC) Hardware Description Language*) e Verilog. Enquanto a primeira suporta tipos de dados abstratos criados pelo próprio utilizador, a segunda linguagem é mais simples e intuitiva, por se assemelhar mais com a linguagem C e, por esse motivo, foi adotada neste trabalho.

De posse dessas ideias, o presente trabalho apresenta a implementação de um sistema compactador de sinais utilizando o dispositivo FPGA e comparando duas metodologias de síntese: a síntese paralela e a síntese em processador dedicado embarcado.

Este artigo está organizado da seguinte forma: a Seção 2 explica o funcionamento da Transformada Wavelet e sua aplicação na compactação de sinais em sistemas elétricos de potência; a Seção 3 apresenta o sistema de medição físico construído no Laboratório de Processamento de Sinais e Telecomunicações

(LAPTEL) na Universidade Federal de Juiz de Fora, bem como as sínteses dos métodos paralelo e com processador embarcado; a Seção 4 mostra os resultados obtidos, comparando os dois métodos. Por fim, a Seção 5 conclui o trabalho e apresenta ideias para estudos futuros.

2 Compactação de Sinais da Rede Elétrica

O sistema elétrico de potência possui, dentre outras características, a propriedade de ser um sistema mais estável quando comparado com outros sistemas menores, ou seja, com menor presença de grandes perturbações. Sua robustez lhe confere um melhor desempenho revelado através de sinais mais comportados sem muitos distúrbios frequentes. Pode-se tirar proveito dessa estabilidade no sentido de se compactar o sinal no seu estado permanente, que oscila na frequência da rede elétrica (60 Hz no Brasil).

Neste trabalho, foi utilizada uma frequência de amostragem de 7680 Hz, o que equivale a uma taxa de amostragem de 128 pontos por ciclo de 60Hz.

Baseado nesses parâmetros, um sistema compactador de sinais elétricos pode ser desenvolvido de tal forma que seja capaz de adquirir as amostras do sinal numa taxa adequada para posterior reconstrução.

A compactação é importante na área de energia elétrica, pois a quantidade de dados armazenados dos registros do comportamento dos sinais é muito grande. Embora o sistema elétrico seja estável, não significa que esteja isento de perturbações ou distúrbios. A qualquer momento pode haver um defeito ou falta no sistema, seja grave ou branda. Por exemplo, um curto-circuito fase-terra ou o acionamento de um motor de indução de grande porte.

O distúrbio modifica o sinal no trecho em que ele ocorre, fazendo com aquele estado seja diferente do estado anterior. Chamaríamos isso de inovação. Portanto, onde existe um distúrbio pode-se dizer que há uma inovação.

Seja qual for a natureza da perturbação, o sistema compactador deve ser capaz de reconstruir todo o tipo de inovação, mesmo sendo de maior frequência. É nesse cenário que a Transformada *Wavelet* se encaixa. Com sua utilização, é possível monitorar tanto as componentes de alta frequência como também as de baixa frequência em tempo real. Isso confere à *wavelet* a capacidade de ser uma ferramenta de análise tempo-frequência.

As componentes de alta frequência geralmente possuem a maioria das amostras com níveis muito baixos de amplitude, podendo ser descartadas quando comparadas a um limite pré estabelecido. Esse descarte causa um erro na reconstrução

que pode ser desprezível quando são comparados os sinais original sem compactação com o reconstruído. Portanto, podemos dizer nesse caso que as componentes de alta frequência são indiferentes à reconstrução, ajudando na compactação.

2.1 Transformada Wavelet

O princípio da Transformada *Wavelet* se dá através de uma decomposição hierárquica de um sinal de entrada em sucessivos componentes de faixas de frequências distintas (Huang et al., 2002), chamados coeficientes da *wavelet*. Através da *wavelet*, podemos analisar o sinal em várias resoluções (Mallat, 1989) e (Galli et al., 1996). Nela, os sinais com altas frequências podem ser vistos com maior amostragem, enquanto os de menores frequências são mais espaçados. Várias aplicações de Transformada *Wavelet* existem, inclusive em outras áreas (Istepanian and Petrosian, 2000).

A implementação da Transformada *Wavelet* é possível utilizando uma cascata de banco de filtros digitais passa-altas e passa-baixas, formando uma espécie de árvore de filtros. Aplicações com banco de filtros podem ser vistas em (Herley and Vetterli, 1993) e (Cohen and Kovacevic, 1996).

Existem vários tipos de *wavelets* utilizadas. Dentre elas, podem-se destacar as do tipo Daubechies. Aplicações desse tipo de *wavelet* podem ser vistas, inclusive para outras áreas em (Michalski and Kacmajor, 2011) e (Elamvazuthi et al., 2013). A *wavelet* utilizada neste trabalho foi a Daubechies3. Esse tipo de *wavelet* foi escolhido de forma empírica, devido ao bom desempenho mostrado na separação das frequências e por sua estrutura não ser tão complexa, facilitando a implementação em *hardware*.

Através da utilização da ferramenta "wfilters" presente no software MATLAB®, podem ser facilmente descobertos os valores dos coeficientes dos filtros digitais a serem cascadeados. São filtros do tipo FIR (*Finite Impulse Response*) de quinta ordem, portanto, representados por 6 coeficientes. Os valores dos coeficientes estão listados na Tabela 1.

A operação de filtragem dos filtros passa-altas (HP) e passa-baixas (LP) são descritas conforme a Equação 1, em que y é a saída do filtro e i é o atraso das amostras do sinal de entrada.

$$y = \sum_{i=0}^5 x[n-i]a_i. \quad (1)$$

Os filtros são conectados em forma de cascata ou árvore. A Figura 1 apresenta um arranjo de banco de filtros utilizados para a implementação da Transformada *Wavelet* desejada. Note que a terminação acinzentada representa um decimador, cujo fator de decimação é 2, na saída do filtro.

Tabela 1: Coeficientes dos filtros passa-altas e passa-baixas.

		Valores em ponto flutuante	Valores em ponto fixo Q15
HP	a_0	-0.3327	-10900
	a_1	0.8069	26440
	a_2	-0.4599	-15069
	a_3	-0.1350	-4424
	a_4	0.0857	2799
	a_5	0.0352	1154
LP	a_0	-0.0352	1154
	a_1	-0.854	-2799
	a_2	-0.1350	-4424
	a_3	0.4599	15069
	a_4	0.8069	26440
	a_5	0.3327	10900

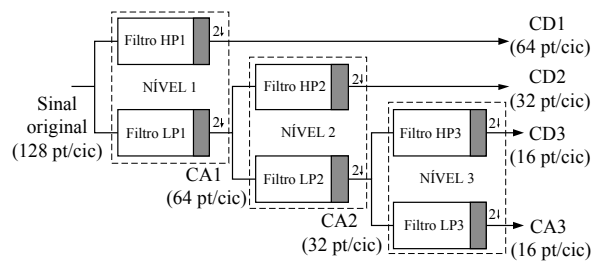


Figura 1: Cascata de filtros formando a *wavelet* composta de três níveis.

Através dessa figura, pode-se ver o arranjo de 3 níveis de bancos de filtros.

O processo começa com a entrada do sinal no primeiro nível. Depois ele é decomposto nos coeficientes de detalhe e de aproximação do primeiro nível (CD1 e CA1, respectivamente). O coeficiente CD1 representa uma saída do processo. CA1 é dividido novamente com filtros iguais aos que existem no primeiro nível, resultando em CD2 e CA2. CD2 se torna outra saída e CA2 é subdividido por fim em CA3 e CD3. Os quatro coeficientes finais utilizados para a etapa de compressão são CD1, CD2, CD3 e CA3. Nesta ordem, esses sinais revelam as componentes de mais alta frequência à mais baixa frequência do sinal original e a partir deles será possível a compactação.

Para reconstruir o sinal a partir dos coeficientes da *wavelet*, basta passá-los por filtros inversos aos da decomposição e então somar os resultados.

A Figura 2 mostra uma representação simplificada da resposta em frequência dos filtros utilizados no esquema. No eixo das abscissas podemos observar a variação da frequência, cujo valor máximo é a metade da frequência de amostragem F_s . O eixo das ordenadas representa o módulo da função de transferência de cada filtro.

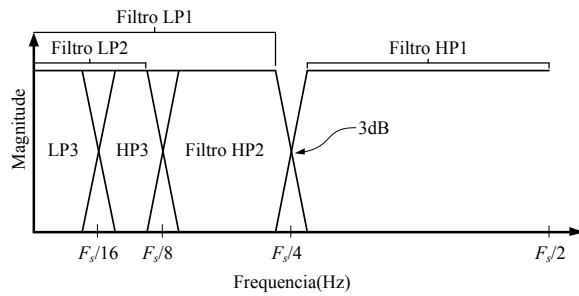


Figura 2: Representação das respostas dos filtros em função da frequência de amostragem.

2.2 Quantização de Coeficientes

O sinal de entrada é amostrado por um conversor analógico-digital de 16 bits de resolução. Optou-se por quantizar os coeficientes dos filtros no mesmo formato, ou seja, o formato Q15. Para isso, basta normalizá-los e multiplicá-los por 2^{15} . Caso ainda existam casas decimais, essas devem ser truncadas. Essa operação resulta na última coluna da Tabela 1.

3 Síntese da Wavelet em FPGA

3.1 FPGA

Dentre vários circuitos integrados de aplicação específica, a FPGA oferece um grande potencial alternativo para acelerar o processo de realização de *hardware* (Huang et al., 2002). A FPGA é uma matriz de milhares de elementos lógicos. Cada elemento lógico é composto, resumidamente, de uma tabela verdade (*look-up table* ou LUT) e um registrador. O resultado da saída de um elemento lógico pode ser ou não registrado, dependendo das ligações entre os blocos. Essas ligações são capazes de "construir" um novo *hardware* dentro da FPGA, que pode ser programada quantas vezes forem necessárias, (Elamvazuthi et al., 2013) e (Fawcett, 1994).

Como o Sistema Elétrico Brasileiro é trifásico, são necessários vários canais de medição para se ter todas as informações das tensões e correntes de todas as fases. Ao todo, podemos contabilizar oito canais, sendo três para as tensões das três fases, outras três para as correntes nas mesmas e os últimos dois canais, para tensão e corrente de neutro.

3.2 Sistema físico

A Figura 3 mostra o *hardware* desenvolvido para a compressão dos sinais. Nesta figura podemos observar os principais componentes/equipamentos utilizados neste projeto.

- i) Gerador de sinais: Gerador da Agilent® com até 15 MHz de frequência do sinal gerado. Tem conexão serial por onde passam os

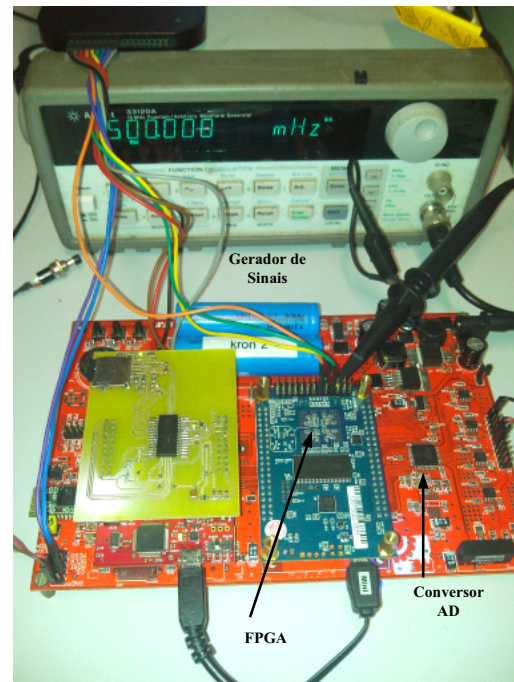


Figura 3: Componentes utilizados.

dados do sinal sintetizado em MATLAB®. O gerador é utilizado para produzir os distúrbios elétricos dos sinais de teste.

- ii) Conversor analógico-digital: O conversor utilizado é o AD7606 da Analog Devices®. Possui 8 canais de conversão simultânea. A resolução é de 16bits. A frequência de amostragem pode chegar a até 200 ksamples/s, embora a utilizada seja de 7680 Hz.
- iii) FPGA: A FPGA utilizada é parte de um kit. O menor kit de desenvolvimento da empresa ALTERA®. O kit é chamado DEONano. Ele Consiste em uma placa pequena que contém o FPGA e mais alguns componentes, tais como memórias, acelerômetros, LEDs, chaves, etc, e o modelo da FPGA é Cyclone 4 - EP4CE22F17C6. Ela possui 22320 elementos lógicos. Além disso, possui também 608 kbits de memória, 154 pinos e 132 multiplicadores de 9 bits embarcados.

Uma das principais etapas do sistema de compactação corresponde a decomposição dos sinais elétricos utilizando *wavelets*. A seguir, serão mostrados dois métodos de síntese: um denominado método paralelo que utiliza blocos convencionais como contadores, multiplicadores, etc, e outro, utilizando um processador embarcado dentro da FPGA, especificamente para fazer os cálculos da Transformada Wavelet.

3.3 Método paralelo

A *wavelet* é formada por seis filtros, todos com o mesmo formato, porém com coeficientes diferentes. Analisando apenas um filtro, é possível entender todo o sistema.

A descrição a seguir, se refere à Figura 4. De acordo com a equação 1, podemos observar o somatório de várias amostras multiplicadas pelos pesos do filtro. A operação do filtro requer que as amostras sejam continuamente atrasadas e multiplicadas novamente pelos pesos do filtro. Para isso, é necessário um atrasador de amostras. Ao serem atrasadas, as amostras vão sendo multiplicadas pelos valores corretos, uma vez que eles estão armazenados numa memória, cujo endereçador é o mesmo contador que seleciona o multiplexador localizado na saída do atrasador. A saída do multiplexador é registrada para chegar no mesmo momento que o coeficiente do filtro e a amostra no multiplicador. Esse, por sua vez, registra e passa ao acumulador. O resultado do acumulador já é o resultado da filtragem.

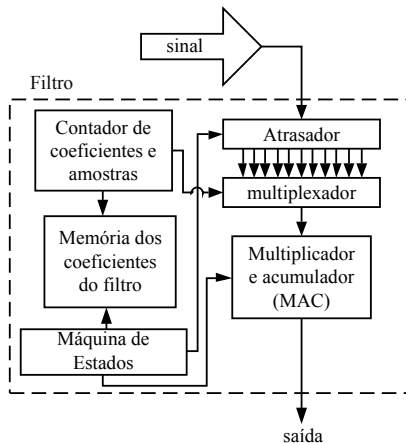


Figura 4: Filtro FIR *wavelet*.

3.4 Processador embarcado

Na seção 3.3 foi descrito o método paralelo para implementação da *wavelet*. Nesta seção é descrito como um processador dedicado pode ser embarcado na FPGA para desenvolver a mesma tarefa. A Figura 5 mostra uma representação do processador com suas conexões internas.

O processador desenvolvido para este trabalho possui arquitetura do tipo Harvard, com instruções RISC (*Reduced Instruction Set Computer*). Os códigos de instrução que vão para a memória dele foram gerados em um compilador C também desenvolvido para esse fim.

Foram utilizadas ferramentas gratuitas da GNU, como o Flex[®] e o Bison[®], ou seja, um código em C foi digitado representando tudo o que os filtros do bloco da *wavelet* fazem de

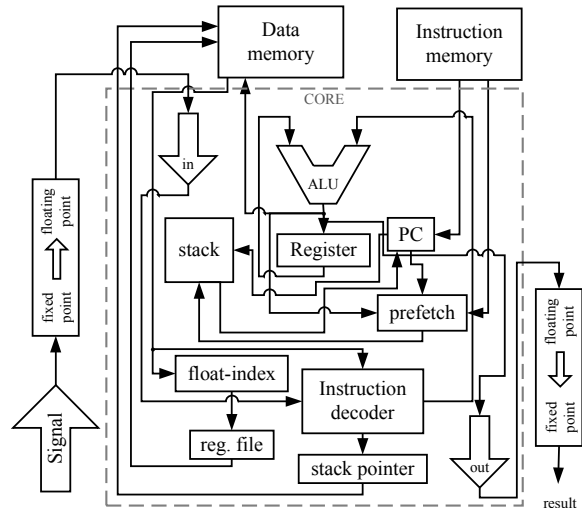


Figura 5: Processador.

operações. Esse código foi compilado e no fim do processo foram criadas instruções binárias para o processador.

O processador trabalha com tamanho de instrução de 14 bits. A memória de instrução possui 512 posições. A memória de dados possui 512 posições de 32 bits. A ULA (Unidade Lógica e Aritmética) possui 64 bits. A vantagem do processador é que ele é desenvolvido para realizar uma tarefa específica, procurando otimizar espaço lógico, podendo trabalhar com altas taxas de clock.

4 Resultados

4.1 Funcionamento do processador

Nesta seção será mostrado o funcionamento do processador embarcado na FPGA. Para isso, o teste foi dividido em algumas etapas. Inicialmente, foi a nível de *software*, através de simulação funcional. Essa simulação é feita a partir de um sinal sintético carregado ao programa. O *software* utilizado para a simulação foi o Modelsim da empresa Mentor Graphics[®]. Com ele, é possível selecionar quais saídas da FPGA serão analisadas.

A Figura 6 mostra a simulação do funcionamento do bloco da *wavelet* implementado na FPGA através do processador embarcado. Analisando essa figura de cima para baixo, os canais no formato analógico são, nesta ordem: O sinal de entrada, em seguida o coeficiente de detalhe CD1, depois o CD2, abaixo deste o CD3, e por fim o CA3. É importante observar o momento do distúrbio transitório, onde as altas frequências também oscilam. É notável também a decimação dos coeficientes de mais alta ordem, dando um aspecto mais quadriculado à onda.

As saídas da simulação foram capturadas e exportadas para o MATLAB[®] para serem

comparadas com a filtragem feita pelo próprio MATLAB® utilizando a ferramenta "filter". Foram então plotadas as saídas para todos os quatro coeficientes a fim de fazer a comparação (Figura 7). Note que o resultado deu o mesmo para a saída do processador e a filtragem no MATLAB®.

Após esse teste, os dados foram gravados em um chip de FPGA e foi feito o teste com sinais elétricos reais provenientes do gerador de sinais. O sinal foi decomposto e reconstruído. Como não foi aplicado nenhum limite nos coeficientes de detalhe, o sinal foi reconstruído perfeitamente. Caso se aplique um limite que zere todos os valores menores que o mesmo, na reconstrução aparecerão pequenas distorções devido a compactação com perda.

A Figura 8 mostra a aplicação do limite nos coeficientes de detalhe CD1, CD2 e CD3 da *wavelet*. O limite usado para a ilustração foi obtido utilizando três vezes o valor do desvio padrão do ruído contido no sinal aplicado ao sistema. Esse valor limite pode ser observado pelas linhas horizontais próximas ao zero. Note que o corte insere zeros nos coeficientes da *wavelet*.

A Figura 9 apresenta a reconstrução de um sinal real proveniente da rede elétrica com o início de uma interrupção. A reconstrução foi feita de duas maneiras, uma sem a aplicação de limite para os coeficientes de detalhe da *wavelet* e outra aplicando-se o limite. Note a diferença entre o sinal original, a reconstrução sem a aplicação do limite e a reconstrução aplicando-se o limite. Esta diferença é muito pequena e aparece só nas áreas de maior turbulência comparando-se o sinal original com a reconstrução onde foi aplicado o limite, ao passo que para o sinal sem o limite aplicado, a diferença é quase nula, podendo ser desprezada.

Ao aplicar esse limite a fim de compactar, alguns parâmetros podem ser avaliados: a quantidade de zeros existentes depois do corte (quanto mais zeros maior será a compactação), a energia mantida e o erro médio quadrático entre a reconstrução com o limite aplicado aos coeficientes de detalhe e o sinal original. A energia E e o erro médio quadrático EMQ podem ser calculados de acordo com as equações 2 e 3, respectivamente, onde N é o tamanho do sinal em questão, $x_{orig}[n]$ são as amostras do sinal original e $x_{rec}[n]$ são as amostras do sinal reconstruído.

$$E = \sum_{n=0}^{N-1} x[n]^2. \quad (2)$$

$$EMQ = \sum_{n=0}^{N-1} (x_{orig}[n] - x_{rec}[n])^2 / N. \quad (3)$$

Para o teste com sinais reais e usando a compactação com perda, houve uma taxa de

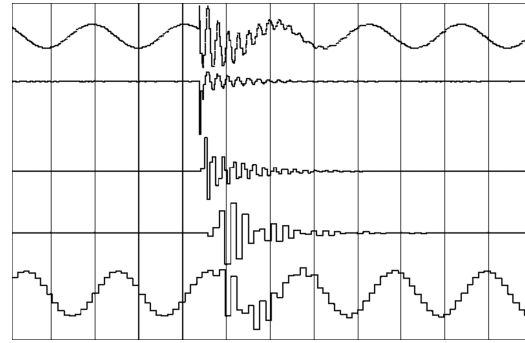
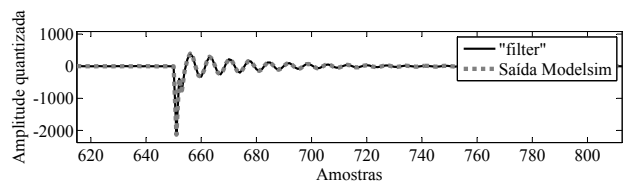
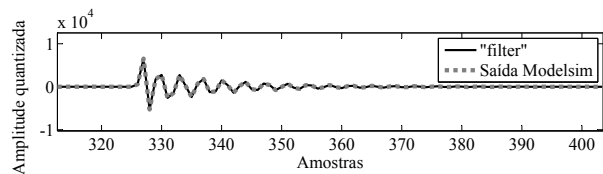


Figura 6: Entrada e coeficientes da *wavelet*. Imagem proveniente do *software* Modelsim.

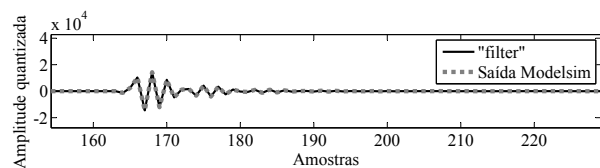
87,42% das amostras substituídas por zero. A energia mantida foi de 99,9352% da energia do sinal original. O erro médio quadrático foi de 0,00024039. Quanto maior o número de zeros, maior é a compactação. Quanto mais próxima de 100% for a energia mantida e quanto menor for o erro, significa que houve menor perda na compactação.



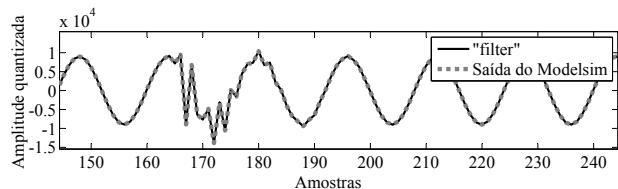
(a) CD1



(b) CD2

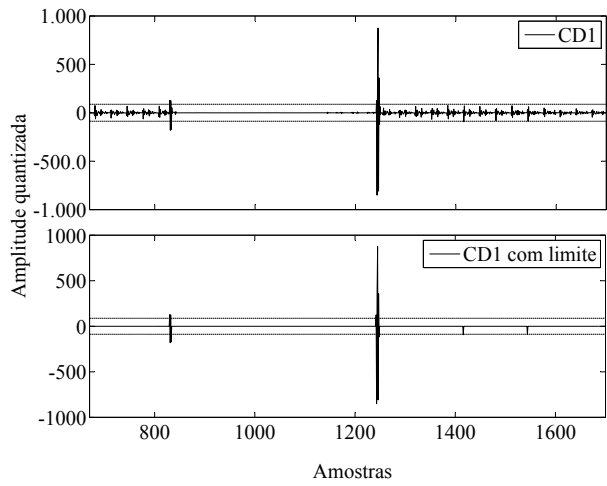


(c) CD3

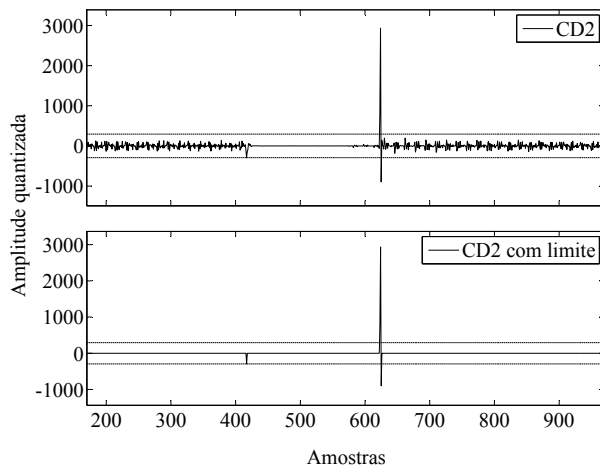


(d) CA3

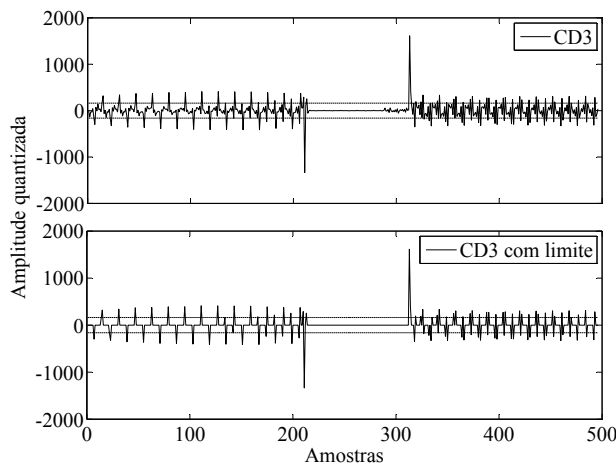
Figura 7: Coeficientes da *wavelet* implementada comparados com o Matlab®.



(a) CD1



(b) CD2



(c) CD3

Figura 8: Aplicação do limite aos coeficientes de detxalhe.

4.2 Comparação entre métodos

A comparação que faremos entre o método paralelo e o processador embarcado reside em

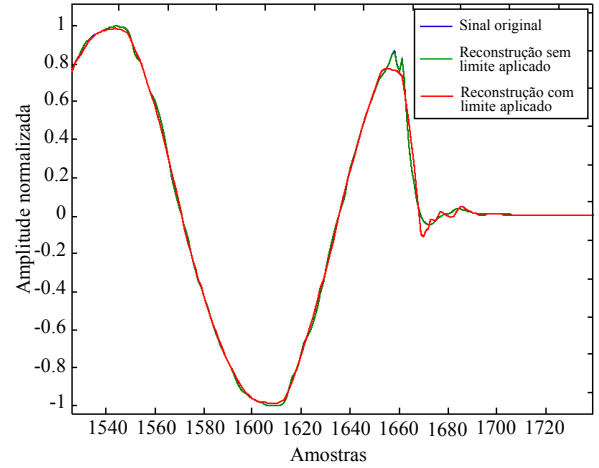


Figura 9: Diferença devido ao limiar.

Tabela 2: Tabela de resultados.

	1c pa	1c pr	8c pa	8c pr
BL	1163	1018	13282	1621
DSP	20	2	132	2
BM(Bits)	0	5074	0	33914
CLK(MHz)	73,46	82,2	75,67	83,96
NCU	50	250	56	1712

termos de lógica utilizada, máxima velocidade de clock permitida e número aproximado de pulsos de clock utilizados, uma vez que ambas as estruturas geram os mesmos resultados numéricos não apresentados neste trabalho.

Para ser feita uma melhor comparação entre os métodos paralelo e com processador, foram feitas quatro implementações cujos resultados estão expressos na tabela 2. A primeira foi realizada com apenas um canal de medição utilizando a implementação paralela, representada pela segunda coluna da tabela (**1c pa**). A segunda implementação foi com um canal utilizando o método do processador embarcado, representada pela terceira coluna (**1c pr**). A quarta e a quinta colunas representam as duas últimas implementações, nas quais foram utilizados os oito canais de medição com o método paralelo e com o processador embarcado, respectivamente.

As linhas da tabela 2 mostram alguns dos recursos utilizados da FPGA: número de blocos lógicos (BL), número de blocos DSP, número de bits de memória utilizados (BM), clock máximo permitido (CLK) e número aproximado de ciclos de clock utilizados (NCU). É importante notar que para uma canal apenas, os dois métodos de implementação são equiparáveis em termos de lógica. Porém, para oito canais, o espaço lógico ocupado pela implementação direta é muito maior, dificultando a inserção de futuros blocos de *hardware*. Conforme cresce o número de canais

de medição, a implementação via processador embarcado consome mais memória da FPGA, pois maior é o número de instruções e dados utilizados nas respectivas memórias.

5 Conclusão

Este trabalho teve como principal objetivo discorrer sobre a implementação da Transformada *Wavelet* em plataforma FPGA, comparando-se dois métodos: o método paralelo e o método com processador embarcado. Pode-se concluir, a partir dos resultados apresentados, que essa ferramenta, principalmente aliada ao processador embarcado em FPGA, representa uma poderosa estratégia para decomposição e compactação de sinais operando em tempo real. Seu emprego permite também a detecção e a análise de distúrbios, sejam eles, ruídos, transitórios ou harmônicos.

A implementação em processador embarcado mostrou-se mais atrativa com relação ao uso de lógica dentro da FPGA, otimizando os recursos utilizados, ao passo que a implementação em paralelo utilizou muito mais blocos lógicos do que a implementação por processador, mas permitiu uma resposta em menos ciclos de clock, o que não é uma forte exigência visto que as amostras são realizadas a uma taxa muito baixa, na ordem de kilohertz.

O estágio atual da implementação do sistema encontra-se operativo para oito canais, contendo além da implementação da *wavelet*, a implementação de outras funções que possibilitam a detecção do evento e a compactação do mesmo. Os próximos passos correspondem à implementação de outros processadores específicos dentro da FPGA, como por exemplo um processador para o cálculo dos componentes harmônicos de acordo com a norma IEC61000-4-7.

Agradecimentos

Os autores gostariam de ao CNPQ a CAPES e a FAPEMIG pelo suporte para a realização desse trabalho.

Referências

- Cohen, A. and Kovacevic, J. (1996). Wavelets: the mathematical background, *Proceedings of the IEEE* **84**(4): 514–522.
- DA SILVA, J. F. (2008). *Sistema de armazenamento de imagens comprimidas através da transformada wavelet*, PhD thesis, Dissertação de Mestrado, Universidade Estadual Paulista Júlio de Mesquita Filho, Ilha Solteira-SP, Brasil.
- de Lima, P. C. (2002). Wavelets: uma introdução, *Matemática Universitária* **33**: 13–44.
- Elamvazuthi, I., Ling, G., Nurhanim, K., Vasant, P. and Parasuraman, S. (2013). Surface electromyography (semg) feature extraction based on daubechies wavelets, *8th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2013, IEEE, Melbourne, Australia, pp. 1492–1495.
- Farge, M., Kevlahan, N., Perrier, V. and Goirand, E. (1996). Wavelets and turbulence, *Proceedings of the IEEE* **84**(4): 639–669.
- Fawcett, B. K. (1994). Tools to speed fpga development, *Spectrum, IEEE* **31**(11): 88–94.
- Galli, A., Heydt, G. and Ribeiro, P. (1996). Exploring the power of wavelet analysis, *IEEE Computer Applications in Power* **9**(4): 37–41.
- Herley, C. and Vetterli, M. (1993). Wavelets and recursive filter banks, *IEEE Transactions on signal Processing* **41**(8): 2536–2556.
- Huang, S.-J., Yang, T.-M. and Huang, J.-T. (2002). Fpga realization of wavelet transform for detection of electric power system disturbances, *IEEE Transactions on Power Delivery* **17**(2): 388–394.
- Istepanian, R. S. and Petrosian, A. A. (2000). Optimal zonal wavelet-based ecg data compression for a mobile telecardiology system, *IEEE Transactions on Information Technology in Biomedicine* **4**(3): 200–211.
- Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**(7): 674–693.
- Michalski, J. J. and Kacmajor, T. (2011). Filter tuning algorithm with compressed reflection characteristic by daubechies d4 wavelet transform, *41st European Microwave Conference (EuMC)*, 2011, IEEE, pp. 778–781.
- Tcheou, M. P., Lovisolo, L., Ribeiro, M. V., da Silva, E. A., Rodrigues, M. A., Romano, J. M. and Diniz, P. S. (n.d.). The compression of electric signal waveforms for smart grids: State of the art and future trends.
- Wagner, V., Balda, J., Griffith, D., McEachern, A., Barnes, T., Hartmann, D., Phileggi, D., Emmanuel, A., Horton, W., Reid, W. et al. (1993). Effects of harmonics on equipment, *IEEE Transactions on Power Delivery* **8**(2): 672–680.