

Performance Evaluation Model based on Precision Reduction and FPGAs applied to Seismic Modeling

Abner C. Barros¹, Bruno H. T. Dutra¹, Vinícius V. Brito¹, Manoel E. Lima¹, Abel Silva-Filho¹,
Rodrigo Gandra² and Ricardo Bragança²

Centro de Informática - Universidade Federal de Pernambuco¹
CENPES/Petrobrás²

{*acb, bhxcd, vvb, mel, agsf*}@*cin.ufpe.br*¹
{*rodrigo.gandra, rbraganca*}@*petrobras.com.br*²

Resumo

The recent increase in computing power of FPGAs has allowed its use in areas such as seismic data processing. Additionally, besides the capability of performing computations in parallel way, FPGAs also support application-specific number representations. In this type of application, in order to achieve better performance, instead of using the floating-point standard, usually the processing and storage of data is done using the fixed point standard. However, the change of representation can cause a degradation in the quality of the results. In the petroleum industry, a seismic image of poor quality can represent an erroneous interpretation of the subsurface, resulting in catastrophic losses. For this reason, it is essential that the quality of data obtained from the seismic data processing for low precision can be evaluated within reliable technical criteria.

In this paper, a real case study was used in order to evaluate the efficiency of two different metrics applied to this seismic application based on RTM algorithm. The main strategy is to explore the precision reduction in terms of SNR (Signal-to-Noise Ratio) and UIQI (Universal Image Quality Index) metrics, in order to improve the performance of the system. Results show a performance gain of 50% compared with the architecture implemented in hardware using floating point standart IEE754.

1. Introdução

Recentemente, a Petrobrás realizou uma das maiores descobertas de Petróleo no Brasil na camada pré-sal com volumes recuperáveis estimados de aproximadamente 10 bilhões de barris de óleo equivalente em Tupi e Guará na bacia de Santos [1], com alta qualidade e conseqüentemente grande valor de mercado.

Atualmente, os algoritmos de imageamento para obtenção de pontos estratégicos de perfuração e extração de petróleo são extremamente complexos e necessitam de alto poder de processamento, podendo levar meses para concluir a simulação completa de uma determinada região. Dentre tais algoritmos de imageamento, o RTM (*Reverse Time Migration*) tem se destacado [2] tendo em vista a qualidade dos resultados que podem ser obtidos mesmo em áreas onde o subsolo possui estruturas muito complexa.

Trabalhos científicos publicados recentemente [3][4][5][7] tem apontado o uso de recursos computacionais não convencionais tais como GPUs e FPGAs na construção de arquiteturas de alto desempenho para o processamento de dados sísmicos.

O uso de tais arquiteturas tem permitido a exploração de estratégias de processamento até então inviáveis nas arquiteturas convencionais, tais como a exploração do paralelismo intrínseco de execução de alguns algoritmos e a mudança de representação dos dados e das operações.

Especificamente no aspecto de mudança de representação dos dados e das operações, redução de precisão é um tipo de abordagem que não pode ser aplicado nas CPUs e nem nas GPUs [6], tendo em vista que estão restritas a um número reduzido de representações tais como inteiro de 32/64-bits e ponto flutuante de simples/dupla precisão. Em arquiteturas tais como FPGAs, é possível representar números reais em hardware utilizando um formato customizado, por exemplo 14 bits, além de permitir um processo de compressão de dados [5], melhorando ainda mais o desempenho da aplicação, principalmente no aumento da largura de banda no acesso a dados na memória do sistema.

Por exemplo, reduzir a precisão de 32-bits ponto flutuante para 16-bits ponto fixo permite melhorar em várias vezes o desempenho da aplicação. No entanto, este tipo de otimização também abre espaço para uma possível degradação de precisão na computação dos

resultados. Com isso, é necessário verificar se a computação acelerada usando precisão reduzida ainda está gerando resultados satisfatórios para um determinado tipo de aplicação.

As contribuições deste artigo estão direcionadas a: (i) desenvolvimento de uma ferramenta que permita explorar o aspecto de redução de precisão na modelagem direta sísmica (ii) avaliação do impacto na qualidade dos resultados gerados com precisão reduzida.

2. Background e Trabalhos Relacionados

Antes de relatar alguns trabalhos relacionados na linha específica de redução de precisão aplicados a arquiteturas não convencionais, será explicado o método de reflexão sísmica atualmente usado pela Petrobrás, que foi usado como base para obtenção de ganhos de desempenho.

O primeiro passo desse método é escolher um local onde será colocada uma fonte de ondas sísmicas. Em seguida, é colocada uma malha de hidrofones que geram sinais elétricos que dependem dos ecos que as ondas sísmicas produzem ao viajar pelo meio. Junto ao hidrofone existe um sistema que faz o armazenamento e visualização dos sinais gerados por esses hidrofones. Os dados armazenados são processados a partir de modelos complexos e os resultados gerados a partir da modelagem sísmica são imagens que servirão como base para que geólogos e geofísicos interpretem o comportamento das ondas sinalizando possíveis pontos estratégicos de perfuração.

A Figura 1 apresenta uma visão geral de como o processo de coleta de dados, através de hidrofones, é realizada[6].

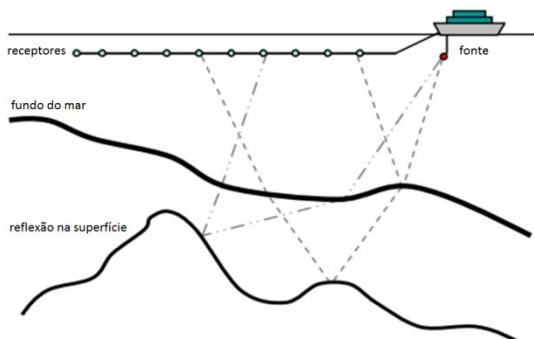


Figura 1 – Visão Geral Método do Reflexão Sísmica[6]

Há algumas técnicas que focam em análise de precisão, muitas das quais são métodos estáticos que operam num fluxo computacional de projeto e usam técnicas baseadas em escala e propagação de erro para realizar a análise.

Lee et al. [8] apresentou uma técnica de análise de precisão estática que usa *arithmetic affine* (onde uma quantidade é representada por uma expressão), para derivar um modelo de erro do projeto e aplica *simulated annealing* para encontrar a mínima largura de bits para satisfazer as restrições de um dado erro.

Uma abordagem similar é mostrada em uma ferramenta de otimização bit-width chamada Preci [9]. Estas abordagens são capazes de realizar uma análise de precisão automática do projeto e provê largura de bits otimizados para as variáveis. No entanto, elas não são completamente adequadas para algoritmos de imageamento sísmico. A primeira razão é que algoritmos de imageamento sísmico usualmente envolvem inúmeras iterações, fato este que pode levar a sobrestimação dos limites do erro e pode derivar uma função de erro sem sentido. Outro aspecto é que a computação em algoritmos sísmicos não tem uma restrição de erro clara. O que pode ser feito é julgar a precisão da computação com base na imagem sísmica gerada.

Haohuan Fu et al. [10] apresentou uma estratégia para determinar a menor precisão aceitável para o processamento de dados sísmicos em FPGAs. Resultados simulados indicam que foi alcançado speedup de 5 a 7 para a abordagem proposta.

Mudança de precisão (quantidade de dígitos do número) e de escala (número de dígitos à direita da casa decimal) são recursos estratégicos para serem tratados em computação de alto desempenho [10]. Para tanto, é preciso um conhecimento aprofundado da aplicação para permitir manipulações no formato do número. Neste artigo, direcionamos nossos esforços na redução de precisão, visando reduzir o tamanho da palavra e permitindo aumentar a largura de banda da memória, com o intuito de buscar mais dados na memória, que é limitada, através de um único acesso. Focamos em 2 tipos de representação de número: ponto fixo e ponto flutuante.

A proposta deste trabalho visa encontrar uma relação adequada de *bit-width* para uma aplicação sísmica. A configuração adequada gerada pela ferramenta foi avaliada na prática através de uma plataforma reconfigurável que será descrita a seguir. Resultados reais de desempenho da aplicação e imagens geradas da modelagem sísmica direta foram avaliados neste trabalho e serão discutidos a seguir.

3. Visão Geral da Arquitetura

O ambiente experimental usado é baseado na plataforma da Gidel [12] que é composta de um FPGA Altera Stratix III 260E [13] conectado a um host Intel Core 2 Quad através de um barramento PCIe x4 e pode

ser ilustrado através da Figura 2. Neste modelo a FPGA funciona como um co-processador de alto desempenho, sendo responsável por executar o processamento dos dados. O *host* é responsável por enviar os parâmetros de configuração, os dados a serem processados para o FPGA, assim como a coleta dos resultados obtidos após a conclusão do processamento.

A placa FPGA é composta de elementos de processamento (PEs) essenciais para maximizar o paralelismo no algoritmo RTM através do método de diferenças finitas, visando uma arquitetura de processamento em *stream* do algoritmo RTM. Esta estrutura compreende um módulo, denominado Arch2D (seria interessante destacar na figura o que seria o Arch2D), responsável por interconectar todos os módulos do FPGA e controlar o fluxo de dados para os núcleos de processamento. O módulo multiport é responsável por operações de escrita e leitura na memória da placa (MEM), além de implementar a comunicação com o barramento PCIe.

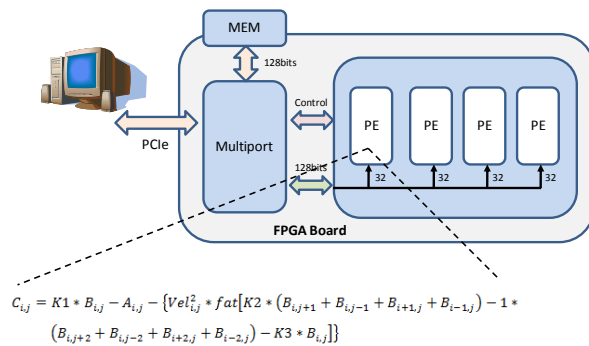


Figura 2 - Visão Geral do Ambiente Experimental

O Núcleo de Processamento (PE) é o módulo da arquitetura para processamento em *stream*, responsável por implementar a equação de diferenças finitas que faz a aproximação da equação de onda acústica que representa a propagação das ondas no meio. Essa equação é utilizada pelo método de migração reversa no tempo. É a resolução dessa equação para cada ponto do modelo, ao longo dos passos de processamentos, que resulta na modelagem sísmica em duas dimensões para o método de migração reversa no tempo. A equação que é implementada pelo PE é mostrada na Figura 2.

A arquitetura descrita nesta seção atingiu um ganho de desempenho em relação ao software de 4.5x, considerando uma máquina Intel Core 2 Quad, 4GB RAM, com compilador otimizado.

Inicialmente, a fim de aumentar o desempenho do sistema, busca-se implementar no FPGA tantos PEs quanto os recursos de hardware disponíveis no

componente permitirem, entretanto, a arquitetura de processamento em *stream* precisa conseguir fornecer dados para os PEs a cada passo de processamento. Neste sentido, reduzir a precisão do PE de 32 bits (caso padrão ponto flutuante) para patamares menores, permitirá inserir mais PEs na arquitetura o que resultará um aumento no desempenho da aplicação. Para tanto, manipular bits no nível de hardware não é uma tarefa trivial, com isso, um modelo de alto nível que permite avaliar redução de precisão de números em ponto fixo/flutuante aplicados ao PE que implementa a equação de modelagem sísmica foi desenvolvido e será detalhado na próxima seção.

4. Modelo Proposto para o Núcleo de Processamento 2D

Esta seção tem como objetivo descrever o modelo parametrizável do núcleo de processamento 2D e todos os seus parâmetros.

O desenvolvimento do modelo tem como objetivo explorar a mudança de representação, de ponto flutuante para ponto fixo dos dados processados pelo PE, assim como a redução de precisão através da diminuição da quantidade de bits da palavra processada pelo PE. No padrão IEEE 754 de ponto flutuante de precisão simples, a palavra possui 32 bits. A idéia do modelo é explorar palavras com menos de 32 bits com intuito de obter ganhos de desempenho na leitura dos dados na memória.

O modelo do núcleo de processamento foi descrito em alto nível utilizando a linguagem de descrição de hardware *SystemVerilog*. A equação do PE descrita na Figura 2 foi dividida em um conjunto de funções com intuito de facilitar o desenvolvimento do modelo. Cada uma das funções do modelo foi transformada em um módulo interno do PE.

Os números das entradas e registradores internos do PE são representados através da notação de ponto fixo. Os números são representados utilizando a representação $Q_{m,n}$, na qual o número possui "*m*" bits na parte inteira e "*n*" na parte fracionária e mais 1 bit de sinal.

O modelo de alto nível do PE foi desenvolvido de forma totalmente parametrizada de forma que a quantidade de bits das entradas e saída de cada função é definida por parâmetros que são configurados automaticamente de acordo com a configuração de três parâmetros principais: o parâmetro NUMBITS que indica a quantidade bits da palavra armazenada no campo de pressão (entradas e saídas do PE); o parâmetro NUMBITS_PINT que indica a quantidade de bits da parte inteira da palavra armazenada no campo; e parâmetro NUMBITS_PFRAC que

representa a quantidade de bits da parte fracionária. Cada parâmetro é configurado em função destes três parâmetros principais com a quantidade de bits necessária para representar o resultado da operação correspondente à função considerada. A configuração $Q_{m,n}$ de cada registrador é ajustada com intuito de preservar o range das operações e fornecer uma precisão aceitável sem que sejam utilizados mais registradores que o necessário para evitar que o PE como um todo ocupe muitos recursos do FPGA. Para evitar, que os registradores internos utilizem muitos bits, os resultados das multiplicações em algumas funções são arredondados. A Figura 4 ilustra uma parte dos parâmetros declarados no modelo e a assinatura de algumas funções.

```

localparam NUMBITS = 32;
localparam NUMBITS_PINT = 8;
localparam NUMBITS_PFRAC= NUMBITS-NUMBITS_PINT-1;
//-----
localparam BITS_M1 = (NUMBITS+2);
localparam BITS_M1_FRAC = NUMBITS_PFRAC;
//-----
localparam BITS_M16 = (NUMBITS+2)+4;
localparam BITS_M16_FRAC = NUMBITS_PFRAC;
//-----
localparam BITS_M60 = (NUMBITS)+6;
localparam BITS_M60_FRAC = NUMBITS_PFRAC;

logic[BITS_P16_SOMA-1:0] M16_soma_r;
logic[BITS_P1_SOMA-1:0] M1_soma_r;
logic[BITS_P60_SOMA-1:0] M60_r;

function logic[BITS_M16-1:0] M16Soma(logic[NUMBITS-1:0] b_i_j_menos_1,
                                     logic[NUMBITS-1:0] b_i_j_mais_1,
                                     logic[NUMBITS-1:0] b_i_menos_1_j,
                                     logic[NUMBITS-1:0] b_i_mais_1_j
                                     );
function logic[BITS_M1-1:0] M1Soma(logic[NUMBITS-1:0] b_i_j_menos_2,
                                   logic[NUMBITS-1:0] b_i_j_mais_2,
                                   logic[NUMBITS-1:0] b_i_menos_2_j,
                                   logic[NUMBITS-1:0] b_i_mais_2_j
                                   );
function logic[BITS_P60-1:0] p60Soma(logic[NUMBITS-1:0] b_i_j);

```

Figura 4 – Parâmetros do modelo e assinatura da funções.

A Figura 5 ilustra o código simplificado do algoritmo RTM, descrito em SystemVerilog alto nível, com o conjunto de funções desenvolvidas no modelo de alto nível do PE executando as operações equivalentes a equação da Figura 2.

Ao configurar o parâmetro NUMBITS_PINT, define-se por sua vez, o range máximo dos valores que serão armazenados no campo de pressão. O range máximo que ocorre no campo de pressão pode variar de modelo para modelo. Desta forma, é preciso definir um range que seja compatível com o modelo utilizado, com o intuito de evitar que, valores mais elevados, que poderão ocorrer no campo de pressão, sejam saturados, e conseqüentemente, haja uma perda de informação

que pode resultar na degradação do resultado final do algoritmo.

A Figura 6 ilustra o comportamento do range no campo de pressão para o modelo de Marmousi [7] no processo de modelagem direta do algoritmo RTM em função dos *timesteps* (passos temporais). O modelo de Marmousi é um modelo utilizado como referência na área de sísmica e representa o modelo de velocidades do terreno.

```

// Loop em T
for (t=0;t<cont_snap+2;++t) begin
...
// Loop em X
for (x=2;x<(nnoib-2;++x) begin
// Loop em Z
for (z=2;z<(nnojz-2;++z) begin
    vv_fat = vel2Fat(VEL[i][j],fat_fixo);
    M16_r = M16Soma(B[i][j-1],B[i][j+1],B[i-1][j],B[i+1][j]);
    M60_r = M60(B[i][j]);
    M1_r = M1Soma(B[i][j-2],B[i][j+2],B[i-2][j],B[i+2][j]);
    M1_16_60_r = MSoma1_16_60(M16_r, M1_r, M60_r);
    MVel2_fatS1_16_60_r = velFatSoma(vv_fat,M1_16_60_r);
    MmenosAij_r = MmenosAij(vel_fat_soma,A[i][j]);
    absixabsj_r = Mabsixabsj(absi[i],absj[j]);
    MmultBorda1_r = MmultBorda1(absixabsj,MmenosAij_r);
    Msoma2Bij_r = Msoma2Bij_r(B[i][j],MmultBorda1_r);
    C[i][j] = MmultBorda2(absixabsj,Msoma2Bij_r);
end
end
end
end

```

Figura 5 – Código simplificado do algoritmo RTM com as funções do modelo de alto nível do PE.

Os resultados indicam que para baixos valores de *timesteps*, as variações no range ainda são altas, no entanto, para valores acima de 1700 *timesteps* o range varia entre 1 e 2. Esse comportamento ocorre porque, no início do processamento, a propagação da onda acústica possui um alta energia, e com o decorrer dos *timesteps* essa energia vai diminuindo a medida que a onda se propaga e reflete nas camadas do modelo.

Uma vez que o range máximo é definido, o número de bits da parte inteira é definida. Após a definição da parte inteira, é preciso definir o número de bits utilizado na parte fracionária da palavra processada pelo PE. A parte fracionária definirá a precisão do PE, e influi diretamente na qualidade do resultado gerado pelo algoritmo. A idéia do modelo parametrizado é fixar a parte inteira da palavra do núcleo de processamento 2D, de maneira a garantir o range dos valores que ocorrem no campo de pressão, assim como permitir o teste de diferentes configurações de bits para a parte fracionária, com intuito de analisar o efeito da perda de precisão na qualidade do resultado final do algoritmo.

O principal objetivo do modelo parametrizado é permitir explorar a redução de precisão, através da diminuição do número de bits da parte fracionária da palavra processada pelo PE. Dessa forma, uma palavra

menor do que 32 bits pode ser obtida, permitindo assim, ler mais dados da memória com a mesma vazão de 128 bits, uma vez que a plataforma adotada possui memória conectada ao FPGA com uma largura de 128 bits, gerando assim ganhos de desempenho para a arquitetura como um todo.

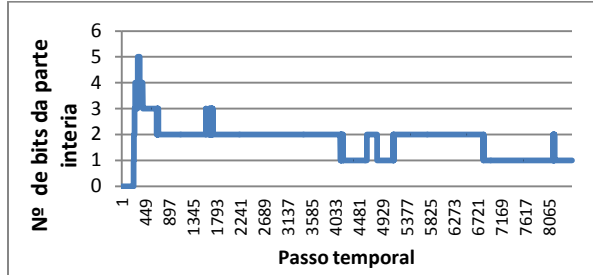


Figura 6 – Range de valores no processamento do Modelo de Marmousi

A qualidade das imagens geradas pelo modelo é avaliada de acordo com uma métrica de avaliação que será explicada na seção 5. As imagens geradas através do modelo são comparadas em relação a uma imagem de referência, utilizando a representação de ponto flutuante gerada pelo o algoritmo em *software*.

5. Métrica para Avaliação dos Resultados Obtidos

A qualidade das imagens geradas no processamento de dados sísmicos é algo extremamente importante. Um erro de interpretação causado por uma imagem de baixa qualidade pode causar prejuízos de milhões de dólares. Desta forma, é essencial que a qualidade dos dados obtidos com o processamento utilizando novas metodologias possam ser avaliadas dentro de critérios técnicos confiáveis.

Conforme foi apresentado na seção 2, atualmente uma das formas encontradas para aumentar o desempenho de sistemas baseados em ASICs e/ou FPGAs, é a substituição do formato de representação e operação dos dados do padrão de ponto flutuante para outro padrão de menor precisão, como, por exemplo, o padrão de ponto fixo. Tal substituição pode ter um impacto positivo, tanto no poder de processamento do sistema, com o aumento da quantidade de elementos de processamento (PEs) por área de hardware, bem como no acesso aos dados, uma vez que dados representados com um menor número de bits exigem um menor *throughput* de memória para serem acessados.

Entretanto, tal substituição pode introduzir um erro nos resultados obtidos. Este erro pode ter origem tanto nas operações aritméticas realizadas quanto no

processo de redução de precisão dos resultados obtidos.

Do ponto de vista do observador, este erro poderá representar desde alterações leves, imperceptíveis, até alterações mais graves que inviabilizem o uso dos dados obtidos. A fim de medir o impacto destas alterações na qualidade dos dados obtidos, em nossos estudos trabalhamos com duas métricas distintas, uma mais objetiva, que mede explicitamente o percentual de erro introduzido nos resultados, e uma mais subjetiva, que mede como este erro afeta a percepção humana da qualidade dos resultados obtidos.

Como primeira métrica, mais objetiva, utilizamos a Relação Sinal Ruído (SNR), representada a seguir através da equação (1). Esta métrica tem a vantagem de expressar corretamente a natureza do erro introduzido com a redução de precisão.

Já a segunda, mais subjetiva, foi adotada uma métrica proposta por Zhou Wang [14], denominada Índice Universal de Qualidade de Imagem (UIQI), representada a seguir através da equação (2). Esta métrica busca avaliar elementos relevantes ao observador humano, tais como: contraste, brilho e artefatos estranhos presentes na imagem. A seguir temos as expressões aritméticas que descrevem estas duas métricas.

$$SNR = 20 \times \log \left(\frac{V_{Rms(x)}}{V_{Rms(x-y)}} \right) \quad (1)$$

$$UIQI = \frac{4 \sigma_{xy} \bar{x} \bar{y}}{(\sigma_x^2 + \sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]} \quad (2)$$

Onde x=conjunto de dados originais, processados e armazenados em float; y=conjunto de dados armazenados e/ou armazenados em ponto fixo; $V_{Rms(x)}$ =energia do sinal original; $V_{Rms(x-y)}$ = energia do ruído introduzido; σ_x e σ_y = respectivamente a variância do sinal original e a variância do sinal de menor precisão; σ_{xy} = matriz de co-variância entre o sinal original e o sinal de menor precisão.

Ambas as técnicas foram aplicadas de igual modo aos resultados obtidos em diversas configurações de precisão de processamento e armazenamento dos dados. Os gráficos das figuras 7 e 8 a seguir ilustram os resultados obtidos:

Como se pode observar pelo gráfico da Figura 7, nas 6 primeiras configurações, entre 10 e 15 bits de precisão, obtivemos uma melhora significativa na relação sinal ruído, SNR, dos resultados obtidos a cada bit de precisão acrescentado. Entretanto, ao contrário do que se poderia esperar, esta medida se estabiliza e não apresenta nenhuma melhora perceptível a partir da configuração com 15 bits de precisão.

Analisando o gráfico da Figura 8, verificamos que, da mesma forma que o SNR, o grau de semelhança apresenta um rápido crescimento entre as configurações de 10 e 15 bits. A partir deste ponto, esta melhora se torna mais discreta a cada bit acrescentado, aproximadamente 0,0056 por bit.

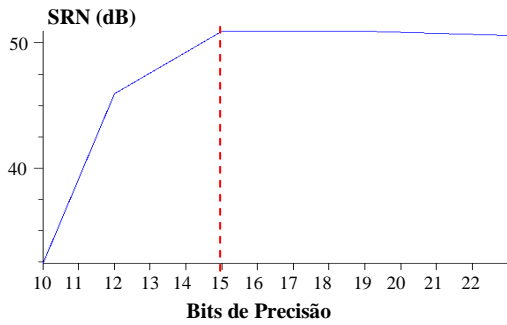


Figura 7 - Resultado obtido com a métrica Índice Universal de Qualidade de Imagem (UIQI)

Figura 8 - Resultado obtido com a métrica Índice Universal de Qualidade de Imagem (UIQI)

Como foi dito no início desta seção, as duas métricas utilizadas tinham objetivos distintos. Enquanto que o SNR buscava medir explicitamente o erro introduzido nos dados devido à redução de precisão, o UIQI buscava medir como este nível de ruído seria percebido por um observador humano.

Observe que em nosso experimento tanto a representação quanto a operação dos dados foi alterada do padrão de ponto flutuante para o padrão de ponto fixo. Desta forma, uma primeira hipótese a ser verificada, a fim de explicar o comportamento observado nos resultados obtidos, seria de que o ruído gerado pela operação em baixa precisão estaria de alguma forma impedindo um melhor resultado nas configurações com mais de 15 bits de precisão, uma vez que configurações com mais bits de precisão deveria produzir resultados mais precisos, com menor nível de erro.

Por este motivo, outro experimento foi realizado com o objetivo de medir o nível de erro causado apenas pela mudança de representação dos resultados.

Através de simulação em software, manteve-se o processamento dos dados em ponto flutuante, convertendo em seguida todos os resultados obtidos para ponto fixo. Os resultados deste experimento podem ser vistos na Figura 9.

Como pode-se observar, desta vez a relação sinal ruído manteve um crescimento aproximadamente constante em todas as configurações, atingindo 110 dB com 19 bits de precisão. Da mesma forma, o grau de semelhança atingiu uma taxa próxima a 90% já com 15 bits de precisão. Comprovando assim a hipótese de que o ruído de operação estava dominando a partir da configuração com 15 bits de precisão. Este efeito estará sendo verificado e melhor estudado em futuros experimentos.

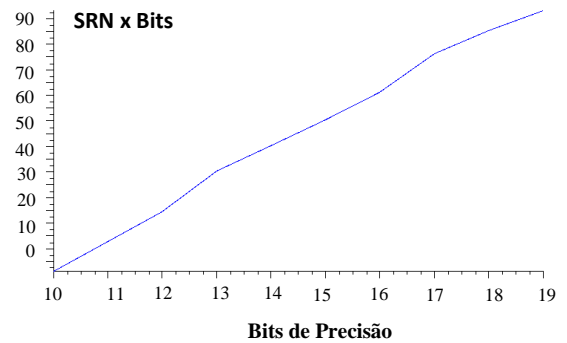
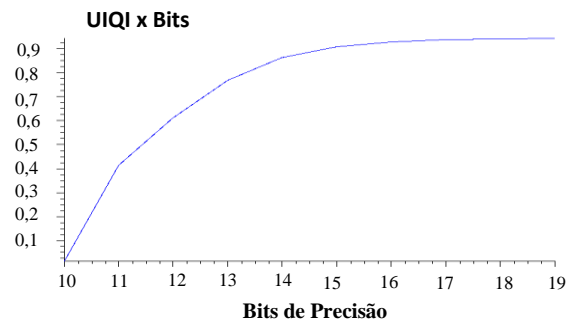


Figura 9 - Resultados obtidos com o processamento em float e o armazenamento em ponto fixo

Outro resultado que chamou a atenção neste experimento foi o fato de novamente a medida UIQI não apresentar melhoras significativas a partir da configuração com 15 bits de precisão. Este resultado, ainda que devendo passar pelo crivo de um especialista em processamento de dados sísmicos, aponta esta como sendo a precisão mínima necessária para o armazenamento de dados de sismogramas.

6. Análise dos Resultados

Os resultados obtidos em nossos experimentos confirmam a viabilidade da utilização de padrões de

representação numérica de menor precisão no armazenamento e na operação de dados sísmicos.

Apesar de a geração de sismogramas sintéticos ser apenas uma parte do algoritmo RTM, o fato de as outras etapas de processamento estarem baseadas na mesma equação apresentada nas figuras 2 e 3, indica que os resultados obtidas neste estudo devem ser igualmente válidas nas outras etapas deste algoritmo.

Os módulos de hardware implementados foram simulados nas seguintes configurações de ponto fixo: 32 bits (8 na parte inteira e 23 na parte fracionária), 25 bits (5 parte inteira e 19 na parte fracionária), 21 bits (5 na parte inteira e 15 na parte fracionária), 18 bits (5 na parte inteira e 12 na parte fracionária) e 16 bits (5 na parte inteira e 10 na parte fracionária). Os resultados obtidos podem ser vistos nas figuras a seguir.

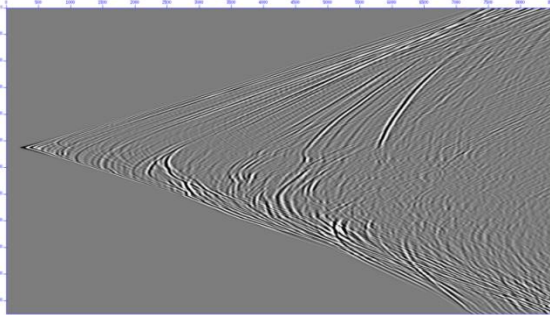


Figura 10 - Sismograma padrão gerado em float

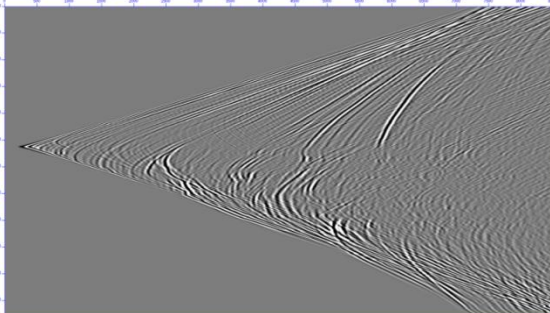


Figura 11 - Sismograma gerado em ponto fixo com 32 bits (8 bits parte inteira, 23 bits de precisão, 1 bit de sinal)

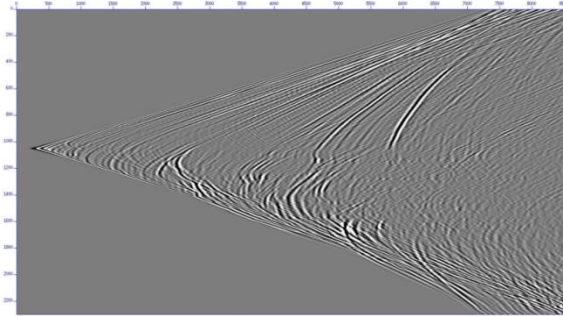


Figura 12 - Sismograma gerado em ponto fixo com 25 bits (5 bits parte inteira, 19 bits de precisão, 1 bit de sinal)

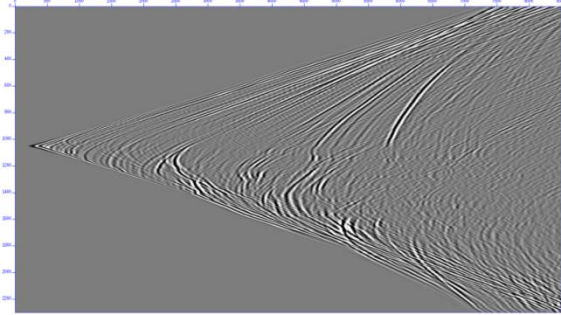


Figura 13 - Sismograma gerado em ponto fixo com 21 bits (5 bits parte inteira, 15 bits de precisão, 1 bit de sinal)

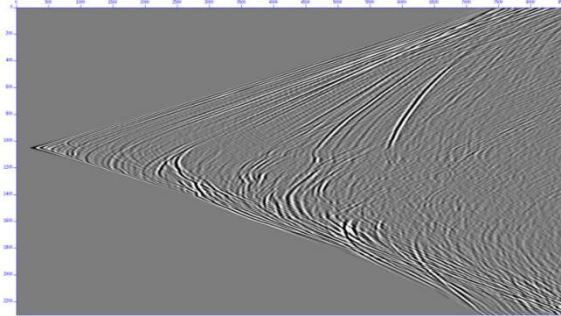


Figura 14 - Sismograma gerado em ponto fixo com 18 bits (5 bits parte inteira, 12 bits de precisão, 1 bit de sinal)

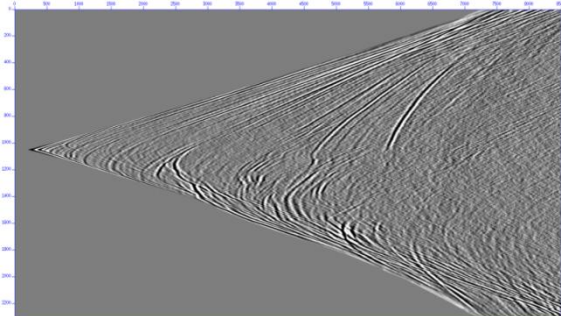


Figura 15 - Sismograma gerado em ponto fixo com 16 bits (5 bits parte inteira, 10 bits de precisão, 1 bit de sinal)

Ao se comparar as imagens obtidas (Figuras 10, 11, 12, 13, 14 e 15) com a imagem de referência em Ponto Flutuante (Figura 10), percebe-se que apenas as imagens de sismogramas com 12 e 10 bits de precisão (Figuras 14 e 15 respectivamente), apresentam diferenças mais pronunciadas. A imagem gerada com 15 bits de precisão (Figura 13) apresenta diferenças sutis, que merecem uma avaliação mais detalhada. As imagens produzidas com 19 e 23 bits de precisão (Figuras 11 e 12 respectivamente) podem ser consideradas idênticas à imagem de referência.

Pelos resultados de síntese obtidos até o presente momento, Tabela 1 a seguir, observa-se que a estratégia da adoção do padrão de ponto fixo permitiu em todas as configurações avaliadas tanto uma redução na área de hardware utilizada quanto um aumento na

freqüência de operação dos Elementos de Processamento (PE) da arquitetura em estudo.

Especificamente no quesito área de hardware, os resultados apontam que seria possível uma melhora de mais de 3x na quantidade de elementos de processamento (PE) do sistema. Entretanto, em nossa implementação, restrições no *throughput* da memória da plataforma utilizada, restringiu em 50% o acréscimo no número dos Elementos de Processamento (PE). Desta forma, tem-se um ganho de desempenho de 50% em relação à arquitetura que utiliza ponto flutuante. Logo o desempenho da arquitetura com 6 PEs passa a ser de 6.75x em relação ao software.

A Tabela 1 a seguir ilustra os resultados de síntese das diversas configurações de PE utilizando ponto fixo e do mesmo PE implementado utilizando ponto flutuante.

Tabela 1 – Resultados de síntese dos diversos PEs utilizando a ferramenta Synplify. O FPGA utilizado foi o Stratix III 260E.

	ALUTS	Registradores	Frequência
PE-Float	6491 (3%)	4260 (2%)	178 MHz
PE-FIXO (32 bits)	1888 (<1%)	1888 (<1%)	226 MHz
PE-FIXO (25 bits)	1464 (<1%)	1435 (<1%)	226 MHz
PE-FIXO (21 bits)	1393 (<1%)	1224 (<1%)	227 MHz

7. Conclusões e Trabalhos Futuros

Os resultados obtidos demonstram as vantagens e os riscos da mudança do padrão de representação de ponto flutuante para ponto fixo em aplicações de processamento de dados sísmicos.

As métricas utilizadas mostraram-se eficientes na verificação da qualidade dos resultados obtidos, atestando a validade da metodologia utilizada, indicando inclusive os valores de grau de semelhança e relação sinal ruído mínimos aceitáveis.

Os resultados de síntese apresentados na Tabela 1 mostram que a arquitetura proposta apontou a possibilidade de ganhos na ordem de mais de 3x no número de Elementos de Processamento e de mais de 27% na freqüência de operação do sistema caso fosse adotado 21 bits de precisão, entretanto limitações no *throughput* da memória na plataforma de hardware utilizada limitaram este a um fator de 50%.

Os resultados obtidos apontam para a necessidade de um estudo mais acurado da fonte do ruído de operação observado no Elemento de Processamento.

8. Agradecimentos

Os autores gostariam de agradecer ao Centro de Pesquisa da Petrobrás (CENPES), à coordenação da

Rede Finep/RPCMod e FACEPE pelo suporte financeiro parcial ao projeto.

9. Referências

- [1] Estatísticas da Petrobrás referentes ao Pré-Sal. Disponível em: <http://www.petrobras.com.br/presal/>. Acessado em Junho de 2011.
- [2] R. G. Clapp, H. Fu, and O. Lindtjorn. "Selecting the right hardware for Reverse Time Migration". In: *The Leading Edge* 29: 48-58, January 2010.
- [3] Thomas, D.B.; Howes, L.; Luk, W. "A Comparison of CPUs, GPUs, FPGAs, and Massively Paralell Processor Arrays for Random Number Generation". *FPGA 2009*, pp.63-72.
- [4] S. Che, J. Li, J.W. Sheaffer, K. Skadron, and J. Lach. "Accelerating Compute Intensive Applications with GPUs and FPGAs". In: *Proc. Symp. Application Specific Processors*, pp. 101-107, 2008.
- [5] M. B. Gokhale and P. S. Graham. "Reconfigurable computing: Accelerating computation with field-programmable gate arrays". Springer-Verlag, ISBN: 0387261052. New York, 2005.
- [6] Jairo Panetta et al. "Computational Characteristics of Production Seismic Migration and its Performance on Novel Processor Architectures", In: (SBAC-PAD), October 2007
- [7] Trevor Irons. Marmousi Model. <<http://www.ahay.org/RSF/book/data/marmousi/paper.pdf>>. Consultado em: 18 de Julho de 2010.
- [8] D.-U. Lee, A. A. Gaffar, R. C. C. Cheung, O. Mencer, W. Luk, and G. A. Constantinides, "Accuracy-guaranteed bit-width optimization," *IEEE TCAD*, vol. 25, no. 10, pp. 1990–2000, 2006.
- [9] G. Shan and B. Biondi, "Imaging steep salt flank with planewave migration in tilted coordinates," *SEG Technical Program Expanded Abstracts*, vol. 25, no. 1, pp. 2372–2376, 2006.
- [10] H. Fu, W. Osborne, R. G. Clapp, O. Mencer and W. Luck. "Accelerating Seismic Computations Using Customized Number Representations of FPGAs" *EURASIP Journal on Embedded Systems*, vol. 2009, Article ID 382983, doi: 10.1155/2009/382983, pp. 1–13, 2009.
- [11] IEEE. Padrão para números em ponto flutuante. Disponível em: <http://grouper.ieee.org/groups/754/>. Acessado em: Junho de 2011.
- [12] Plataformas da GiDEL: PROCe III. Disponível em: <http://www.gidel.com/PROCe%20III.htm>. Acessado em Junho de 2011.
- [13] FPGA Stratix III da Altera. Disponível em: <http://www.altera.com/products/devices/stratix-fpgas/stratix-iii/overview/st3-overview.html>. Acessado em Junho de 2011.
- [14] Zhou Wang, Alan C. Bovik, "A Universal Image Quality Index", *IEEE SIGNAL PROCESSING LETTERS*, vol. 9, Vol. 3, 2002.