

DECODIFICADOR DE ÁUDIO MPEG-2 AAC-LC: SOMADORES OTIMIZADOS EM PONTO FLUTUANTE DO MÓDULO IMDCT

Karine Souza de Almeida¹; Wagner Luiz Alves de Oliveira.²

1. Bolsista PIBITI, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: karine.almeida@gmail.com
2. Orientador, Departamento de Tecnologia, Universidade Estadual de Feira de Santana, e-mail: wlaol@uol.com.br

PALAVRAS-CHAVE: IP-Cores, Circuitos Integrados, Aritmética de Ponto Flutuante.

INTRODUÇÃO

Notoriamente, diversas aplicações utilizam números reais para realizar o processamento de informações. E para isto, são necessárias ferramentas computacionais que consigam operar esse conjunto de valores. Tratando-se de aritmética computacional, é preciso representar esses valores no sistema de numeração binária, onde para representar números inteiros muito grandes, são necessários muitos bits. O sistema em ponto flutuante baseado em notação científica é capaz de representar números muito grandes e muito pequenos sem aumentar o número de bits e também representa números que possuem parte inteira e fracionária (FLOYD, 2006).

Esse projeto visa a implementação de unidades somadoras em ponto flutuante para o bloco IMDCT do Decodificador de Áudio MPEG-2 AAC-LC, o qual está sendo desenvolvido na UEFS para fins acadêmicos, podendo ser utilizado futuramente para fins comerciais em sistemas digitais.

MATERIAIS E MÉTODOS

Em relação à parte de estudos teóricos, para o referente projeto foram consultados:

- 1) Trabalho de Conclusão de Curso de João Marcelo Silva Souza, estudante de graduação da UEFS, orientado pelo prof. Dr. Wagner Luiz de Oliveira, do Departamento de Tecnologia da UEFS.
Este trabalho trata do desenvolvimento e implementação em lógica digital da função de janelamento KBD do módulo **Windowing/Block Switching** de um Decodificador de Áudio MPEG-2 AAC-LC. Este projeto aborda conceitos relacionados à aritmética de ponto flutuante, ditando a norma IEEE-754.
- 2) Projeto Anual dos professores Tiago Oliveira e Delmar Bróglia Carvalho, do Departamento de Tecnologia da UEFS.
Este projeto apresenta o desenvolvimento de uma unidade aritmética de ponto flutuante. As seções requeridas para o desenvolvimento do projeto são as que tratam de operações aritméticas em ponto flutuante e implementação de somadores usando

Verilog (Linguagem de Descrição de Hardware), mostrando alguns possíveis algoritmos analisados no corrente projeto.

A metodologia de projeto ipProcess, utilizada pelo Brazil-IP (BARROS, 2005), também foi estudada. O Brazil-IP é um consórcio de universidades brasileiras, cujo objetivo é desenvolver mão-de-obra qualificada, em nível de graduação, no projeto de circuitos integrados. Para tal, este consórcio ampara-se fortemente na metodologia ipProcess, voltada para a prototipação em FPGA de sistemas digitais, de forma a validar a especificação dos mesmos, para adequação ao fluxo ASIC (*Application-Specific Integrated Circuit* – Circuito Integrado de Aplicação Específica).

O projeto Brazil-IP ampara-se fortemente no processo de verificação funcional (BERGERON, 2003 & MELCHER, 2008), como forma de garantir o correto funcionamento do circuito integrado, antes do envio do arquivo GDSII para a foundry (a fábrica que produzirá o circuito integrado). Neste sentido, a bolsista participou de um intenso treinamento de 40h, provido por um pesquisador da UFCG, vinculado ao Brazil-IP, sobre verificação funcional.

RESULTADOS

Existem métodos utilizados por alguns somadores para realizar a operação de soma, cada um apresentando suas características em termos de tempo de resposta e custo. O *Ripple Carry* é um somador caracterizado pela implementação simplificada de seu algoritmo. O *Carry Save* realiza soma de mais de dois números ao mesmo tempo. O *Carry Select* é um somador que é dividido em setores e executa duas somas em paralelo. O somador *Carry Look-Ahead* possui esquemas para antecipar o *carry*. O estudo sobre esses somadores é fundamental para a escolha de um algoritmo que atenda os requisitos da unidade de ponto flutuante desenvolvida para o bloco **Windowing/Block Swtching**.

Durante o desenvolvimento do projeto houve um remanejamento de atividades e, ao invés de desenvolver as unidades somadoras em ponto flutuante, integrando-as ao **IMDCT** do **Decodificador de Áudio MPEG-2 AAC-LC**, as mesmas começaram a se desenvolver para o módulo **Windowing/Block Switching** do decodificador. Esse módulo é responsável por realizar um processo de janelamento (etapa no processo de reconstrução do sinal de áudio) entre as amostras de áudio e entregar as amostras janeladas na saída. Tais amostras necessitam ser armazenadas na memória. Mas para realizar esses passos é necessário gerenciar os acessos à memória para garantir que as amostras sejam armazenadas na mesma ordem com que foram obtidas durante o processo de amostragem e retenção. O controle de acesso à memória garante que a mensagem seja guardada de acordo com a mensagem original de áudio que foi passada. Então, para esse fim, um importante módulo controlador foi desenvolvido e implementado.

O módulo controlador gerencia os acessos à memória realizados pelos módulos IMDCT e Overlapping/Adding. A IMDCT gera 2048 endereços de memória, cada um de 11 bits, para realizar a busca de dados, de 16 bits cada, em uma memória de 2048 posições. Os dados armazenados na memória representam, como já dito, as amostras do sinal de áudio e são passadas para uma fila (onde o primeiro dado a entrar deve ser o primeiro a sair - FIFO), onde são solicitados pelo **Windowing/Block Switching**. Em seguida, esses dados são janelados através da função Seno ou KBD, colocados em outra estrutura de dados (outra fila) e posteriormente solicitados pelo Overlapping/Adding, onde ocorre a parte final da reconstrução do sinal.

Os sinais de entrada da IMDCT são: **Full**, **BUS_Granted** e **BUS_Assigned** e **w_sequence Full**. E Os sinais de saída são: **Bus_Request**, **BUS_Control_BUSY** e **Address**.

Um sinal **Full** vindo da FIFO indica se a fila está cheia de dados ou não. Caso **Full** assuma valor 0 a FIFO não está cheia e é possível fazer a solicitação ao controlador para utilizar o barramento de endereço. Com **BUS_Request** a IMDCT solicita o barramento ao controlador. Caso o Overlapping/Adding não tenha solicitado o barramento primeiro, a permissão é concedida à IMDCT. **BUS_Granted** e **BUS_Assigned** vêm do controlador e quando estão em 1 indicam que a IMDCT pode utilizar o barramento para enviar endereços (**Address**) para realizar acessos a dados na memória. E então a IMDCT devolve uma resposta ao controlador confirmando que vai utilizar o barramento com o **BUS_Control_BUSY**. O **w_sequence** é a entrada vai indicar qual o tipo de sequência de janela que deve ser aplicada ao sinal. A partir do valor de **w_sequence** é define-se como será o acesso à memória. Esse acesso pode acontecer de forma linear ou não linear, a depender do tipo de sequência lida na entrada da IMDCT. Caso a sequência seja **ONLY_LONG_SEQUENCE** - 0, **LONG_START_SEQUENCE** - 1 ou **LONG_STOP_SEQUENCE** - 3, os endereços serão gerados pela IMDCT de forma linear, que por consequência serão realizados acessos à memória do modo linear também. Porém, se a sequência for **EIGHT_SHORT_SEQUENCE**, então os endereços gerados pela IMDCT não serão linear, mas alternados, gerando por consequência um acesso não linear à memória. A Figura 1 mostra uma simulação da IMDCT utilizando a ferramenta QUARTUS II.

O Overlapping/Adding funciona de maneira semelhante à IMDCT, diferenciando-se apenas que, ao invés de gerar endereços para a partir deles ler dados da memória, esses endereços serão utilizados para realizar escrita de dados na memória enquanto a FIFO não estiver vazia.

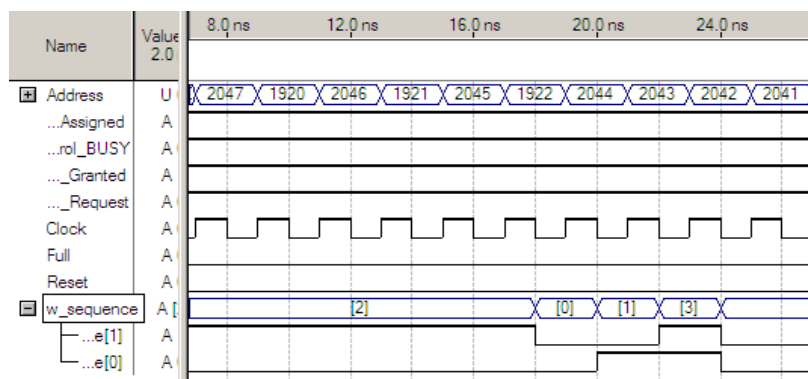


Figura 1. Formas de endereçamento realizado pela IMDCT

CONSIDERAÇÕES FINAIS

Diante dos resultados aqui apresentados, percebe-se que este trabalho contribuiu para o desenvolvimento de algo maior, o Decodificador de Áudio MPEG-2 AAC- LC do programa Brazil-IP/UEFS.

Como o projeto teve que ser modificado durante o processo de desenvolvimento para a implementação dos módulos **IMDCT** e **Overlapping/Adding**, foram encontrados alguns tipos de problemas, tais como o endereçamento correto das amostras vindas do **Windowing/Block Switching**, a qual exigiu acesso e gravação na memória de maneira linear em alguns momentos e de maneira não linear em outros. Além disso, fez-se necessário encontrar alternativas de algoritmos computacionais que reduzissem o espaço de armazenamento e o tempo de acesso ao mesmo.

REFERÊNCIAS

- BARROS, E. et al. iprocess: Using a process to teach ip-core development. Microelectronics Systems Education, IEEE International Conference on/Multimedia Software Engineering, International Symposium on, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 27-28, 2005.
- BERGERON, J. Writing Testbenches using System Verilog. 233 Spring Street, New York, NY 10013, USA: Springer, 2003.
- BRAZIL-IP. Rede brasileira de centros de concepção de Sistemas Digitais e IP-Cores. 2002. Disponível em: <<http://www.brazilip.org.br/>> 07 jul. 2011.
- FLOYD, Thomas L. Digital Fundamentals. 9^a ed. New Jersey, Prentice Hall, 2006.
- ISO/IEC 13818-7 Information technology, Generic coding of moving pictures and associated audio information - Part 7: Advanced Audio Coding (AAC), Third edition, 2004.
- MELCHER, E. U. K. Verificação funcional. Curso do programa Brazil-IP. 2008. Universidade Federal de Campina Grande.