DESCRIÇÃO EM VHDL DE ALGORITMOS QUÂNTICOS

PEDRO HENRIQUE G. MARCHI[1],
RENATA H. S. REISER[1], MAURÍCIO L. PILLA[1]

[1]CDTEC – Universidade Federal de Pelotas (UFPEL) – Pelotas – RS – Brasil - {phgmarchi,reiser,pilla}@inf.ufpel.edu.br

1. INTRODUÇÃO

Relevantes aplicações do VHDL (*VHSIC Hardware Description Language)* (PEDRONI, 2004) estão no campo da programação de dispositivos lógicos como FPGA e das aplicações específicas de circuitos integrados ASICs (*Application Specific Integrated Circuits*).

A geração do código VHDL pela ferramenta Quartus II (ALTERA, 2007) desenvolvida pela empresa Altera viabiliza o projetado e simulação de FPGA (*Field Programmable Gate Array*), validando o código desenvolvido. Dentre suas principais características, destaca-se uma grande potencialidade na interpretação da simulação através de suas diversas ferramentas, viabilizando a análise da complexidade (números de bits, portas) e a correspondente dimensão da área do circuito interno gerado. Ao contrário da programação essencialmente baseada em computações sequenciais, no VHDL a programação é concorrente. Ao evitar a limitação gerada pelo acesso à memória sequencial, torna-se uma linguagem atrativa para execução de algoritmos paralelos. Pela geração do código VHDL e aplicação de FPGAs, tem-se uma opção para gerar um aumento significativo na velocidade e no número de qubits quando da simulação quântica, fornecendo uma alternativa de análise e desenvolvimento de algoritmos quânticos.

Assim, a motivação para uso de VHDL neste trabalho é buscar uma descrição dos circuitos quânticos a partir dos padrões estabelecidos para os circuitos clássicos. A principal contribuição na atual etapa de trabalho, é viabilizar a simulação do paralelismo quântico associada às portas unitárias que compõem um conjunto universal para manipulação de circuitos quânticos. Em particular, uma biblioteca de métodos para especificação de coeficientes, números complexos normalizados está em desenvolvimento, incluindo operações aritméticas e considerando a representação polar, baseada no módulo e ângulo.

O principal objetivo é o desenvolvimento de uma extensão da biblioteca qEx-VHDL, pela construção de um módulo qEx-VHDL-FP, contendo funções específicas para manipulação de qubits e portas lógicas quânticas unidimensionais.

2. METODOLOGIA

Em analogia a construção clássica, uma computação quântica pode ser traduzida em termos de portas lógicas quânticas, as quais são realizadas por

elementos conectados entre si, de tal forma a obter o sinal de saída (estado final) a partir de relações lógicas entre o sinal de entrada (estado inicial).

Se um bit de informação pode armazenar um dos dois valores 0 ou 1, um registrador clássico de n bits pode armazenar um conjunto de 2n elementos ({0, 1, ..., 2n-1}) por vez. No computador quântico, a unidade básica de informação é o qubit, interpretado por um vetor bidimensional do espaço de Hilbert, o qual é definido como um espaço vetorial complexo munido do produto interno (NIELSEN E CHUANG, 2000).

Além de armazenar 0 ou 1, o qubit também armazena a superposição de ambos 0 e 1. Assim, tem-se a representação de um qubit na representação de Dirac:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle ,$$

onde α e β são as amplitudes dos estados básicos, números complexos normalizados ($\alpha^2 + \beta^2 = 1$). Mais especificamente,

$$\alpha = (\rho_\alpha, \theta_\alpha) = |\rho_\alpha|(\cos\theta_\alpha + i\ sen\ \theta_\alpha)$$
$$\beta = (\rho_\beta, \theta_\beta) = |\rho_\alpha|(\cos\theta_\alpha + i\ sen\ \theta_\alpha)$$

são a representação polar definida pelo módulo (ρ) e ângulo (θ) dos respectivos coeficientes (números complexos) do qubit $|\Psi\rangle$.

Por consequência, a informação armazenada em um registrador de n qubits é de 2^n valores, simultaneamente. Ou seja, se um registrador quântico de n qubits pode armazenar 2^n números ao mesmo tempo, um computador quântico deverá processar toda esta entrada de dados, simultaneamente, caracterizando-se assim o paralelismo quântico [Imre e Balázs 2006].

A transformação dos possíveis estados de registradores quânticos pode ser modelada por operadores unitários, referidos como portas quânticas. Tais dispositivos quânticos podem executar uma operação unitária fixada, sobre qubits selecionados, em período determinado no tempo.

Um operador unitário U é interpretado como uma matriz unitária U. Operadores quânticos de um qubit são denominadas portas elementares. Outras operações não unitárias são operadores de controle e medida (neste caso, interpretadas por projeções do vetor correspondente ao qubit sobre um par de subespaços ortogonais) (NIELSEN E CHUANG, 2000).

2.1. METODOLOGIA PARA DESENVOLVIMENTO DO MÓDULO qEX-VHDL-FP

Para a realização dos cálculos, consideramos números complexos α e β na notação polar. Essa notação reduz significativamente o número de cálculos em algumas operações, se comparada com a notação algébrica.

Nas operações de multiplicação e exponenciação, tem-se:

$$\alpha\beta = |\rho_\alpha||\rho_\beta|(\cos\ (\theta_\alpha + \theta_\beta) + i\ sen\ (\theta_\alpha + \theta_\beta));$$

$$\alpha^n = |\rho_\alpha|^n(\cos\ (n\theta_\alpha) + i\ sen\ (n\theta_\alpha)).$$

Nesta representação são realizados cálculos apenas sobre os valores do módulo e do ângulo de cada um dos complexos.

Estas vantagens que se aplicam na etapa de modelagem dos dados também são estendidas na especificação em VHDL, pois é possível fazer uma descrição simplificada, e consequentemente, a geração de hardware mais simples para executar as operações (aritméticas).

3. RESULTADOS E DISCUSSÃO

Na Figura 1, apresenta-se o trecho de código de especificação de operações aritméticas sobre números complexos. Como é possível observar, os cálculos envolvendo os ângulos foram condensados em apenas um operador, já que manipulam os mesmos valores.

A implementação dos operandos utiliza um Sistema de Ponto Flutuante, com precisão de 32 bits, 1 bit de sinal, expoentes de 8 bits e mantissa de 23 bits.

```
entity multiplica is
port (
        modulo1, cos1, modulo2, cos2: in std_logic_vector (31 downto 0);
        mods, coss: out std_logic_vector (31 downto 0)
);
end multiplica;
architecture arc_multiplica of multiplica is
begin
    process (modulo1, cos1, modulo2, cos2)
        variable modtemps, costemps: std_logic;
        variable modtempexp, costempexp: std_logic_vector (7 downto 0);
        variable costempman, sentempman: std_logic_vector (22 downto 0);
        variable mancos1, mancos2: std_logic_vector (22 downto 0);
        variable modtempman: std_logic_vector (45 downto 0);
        variable expcos1, expcos2: std_logic_vector (7 downto 0);
        variable temp: std_logic_vector (6 downto 0);
        variable temp2: integer;
        begin
        modtemps:= (modulo1 (31)) xor (modulo2 (31));
        if (modulo1(30) = '0' and modulo2(30) ='1') then
            if (modulo1(29 downto 23) > modulo2(29 downto 23))then
                modtempexp:= '0' & (modulo1(29 downto 23) - modulo2(29 downto 23));
            end if;
            if (modulo1(29 downto 23) < modulo2(29 downto 23))then
                modtempexp:= '1' & (modulo2(29 downto 23) - modulo1(29 downto 23));
            end if;
            if (modulo1(29 downto 23) = modulo2(29 downto 23))then
                modtempexp:="00000000";
            end if;
        elsif (modulo1(30) = '1' and modulo2(30) ='0') then
            if (modulo1(29 downto 23) > modulo2(29 downto 23))then
                modtempexp:= '1' & (modulo1(29 downto 23) - modulo2(29 downto 23));
            end if;
            if (modulo1(29 downto 23) < modulo2(29 downto 23))then
                modtempexp:= '0' & (modulo2(29 downto 23) - modulo1(29 downto 23));
            end if;
            if (modulo1(29 downto 23) = modulo2(29 downto 23))then
                modtempexp:="00000000";
            end if;
        else
        modtempexp := modulo1(30) & (modulo1(29 downto 23) + modulo2(29 downto 23));
        end if;
        modtempman:= (modulo1 (22 downto 0)) * (modulo2 (22 downto 0));
```

Figura 1. Codificação em VHDL da multiplicação de números complexos.

Na continuidade, tem-se a implementação das demais operações (radiciação, soma, diferença, normalização) incluindo ainda os operadores unitários (transformações unitárias, controladas e medidas).

Além destes, considera-se também a otimização da implementação bem como a simulação em diferentes configurações de FPGAs para análise de desempenho e tamanho dos circuitos quânticos simulados em VHDL. Aplicações específicas para algoritmos básicos da computação estão em desenvolvimento.

4. CONCLUSÕES

Este artigo teve como objetivo geral difundir os conceitos, tecnologias e aplicações da simulação quântica via *hardware* enquanto área estratégica para o desenvolvimento científico e tecnológico. A consolidação da biblioteca qEx-VHDL-FP considera uma análise teórica e formal das propriedades especiais dos qubits (paralelismo quântico, superposição e emaranhamento) ao nível do modelo de circuitos quânticos para estudo/reformulação de noções em diferentes níveis de abstração usuais em sistemas computacionais.

A construção de métodos para manipulação dos coeficientes dos qubits é uma das principais etapas na concepção e implementação de métodos específicos para simulação de algoritmos quânticos baseada no uso de linguagem de descrição de *hardware* VHDL e, na execução de circuitos quânticos via *hardware*, pela utilização de dispositivos lógicos programáveis.

5. REFERÊNCIAS BIBLIOGRÁFICAS

Altera (2007). **Quartus II Version 7.2 Handbook**, volume I. Altera Inc.

IMRE, S. AND BALÁZS, F. **Quantum Computing and Communications an Engineering Approach**. John Wiley & Sons, NJ, 2005.

MONTEIRO, E. R., JACCOTTET, D. P., REISER, R. H. S., COSTA, E., PILLA, M. L. Simulação Quântica em VHDL: Um Estudo de Caso Baseado no Algoritmo de Deutsch. In: **XXXV CONFERÊNCIA LATINO AMERICANA DE INFORMÁTICA**. Pelotas, 2009.

MONTEIRO, E. R., JACCOTTET, D. P., REISER, R. H. S., COSTA, E., PILLA, M. L. Simulação Quântica em VHDL: Um Estudo de Caso Baseado no Algoritmo Quântico de Grover. In: **WSCAD-SSC 2009: X SIMPÓSIO EM SISTEMAS COMPUTACIONAIS WORKSHOP DE INICIAÇÃO CIENTÍFICA**. São Paulo, 2009.

NIELSEN, M. A. AND CHUANG, I. L. Quantum Computation and Quantum Information. **Cambridge University Press**. 2000

Pedroni, V. A. (2004). Circuit Design with VHDL. **MIT Press**. 2004.