

qExVHDL, uma nova abordagem a computação quântica simulada em hardware

Lucas Agostini¹, Julio Machado¹, Renata Reiser¹;

¹Universidade Federal de Pelotas –{lbagostini,jmndsneto,abdavilar,reiser,pilla}@inf.ufpel.edu.br

1. INTRODUÇÃO

Este trabalho estende as atividades do Projeto qExVHDL~\cite{Monteiro}, o qual considera o uso da linguagem VHDL (*VHSIC - Very High Speed Integrated Circuit Hardware Description Language*)~\cite{tutorial} para simulação de algoritmos quânticos (AQs) e correspondente prototipação em FPGA (*Field-Programmable Gate Array*), objetivando uma descrição dos circuitos quânticos (CQs) a partir dos padrões estabelecidos para os circuitos clássicos (CCs).

A simulação via hardware de AQs é uma área de grande interesse para a computação quântica possibilitando análise das propriedades como paralelismo quântico, emaranhamento e superposição de estados (CONCEIÇÃO, 2002). Consolida-se como uma metodologia amplamente usada para o desenvolvimento e depuração de CQs, mas que utiliza massivamente operações de ponto flutuante. Nesta etapa do Projeto qExVHDL, não se aplicam as rotinas em (BISHOP, 2008) para manipulação de ponto flutuante, e se propõe a simulação de CQs através do desenvolvimento de um biblioteca em VHDL sintetizável, incluindo as operações necessárias para o cálculo e simulação de dados clássicos e quânticos, preservando o padrão IEEE. O objetivo da biblioteca é permitir a síntese dos métodos em FPGAs.

2. METODOLOGIA

O modelo de CQs constitui-se numa linguagem universal para descrição de computadores quânticos em aplicações práticas para análise dos atuais algoritmos quânticos. Considerando a notação de Dirac (Nielsen; Chuang, 2000) do q-bit de coeficientes complexos c_0 e c_1 , $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$, seguem-se as convenções usadas nos CCs:

1. *entrada*, indicada um registrador de múltiplos q-bits (geralmente clássicos: $|0\rangle$ ou $|1\rangle$);
2. *linhas horizontais*, representando a evolução temporal (da esquerda para a direita) do fluxo de dados (q-bits) entre duas portas lógicas;
3. *linhas verticais*, mostrando a atuação síncrona em dois ou mais q-bits;
4. *controle*, representado por um círculo no q-bit controle;
5. *portas lógicas*, indicadas por caixas com uma letra de identificação;
6. *medida*, de um q-bit, geralmente ocorrendo na saída do circuito e alterando o estado do q-bit, o qual retorna 0 e 1 com uma distribuição de probabilidade associada;
7. *estados* ϕ_i como transformações em instantes de tempo (t_i).

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 n elementos

por vez (0, 1, ..., n - 1). No computador quântico, um q-bit de informação além de poder armazenar 0 ou 1 também pode armazenar ambos 0 e 1, em superposição. Por consequência, um registrador quântico de n q-bits pode armazenar 2^n bits, simultaneamente, caracterizando o paralelismo quântico.

A evolução dos CQs é modelada por transformações quânticas (TQs) sobre possíveis estados modelados por registradores quânticos, referidos com operadores unitários (dispositivos que executam uma operação unitária fixada, sobre q-bits selecionados, em período determinado no tempo).

CQs são definidos considerando-se apenas conjuntos finitos de portas quânticas elementares (portas quânticas de um q-bit), neste caso, denominados conjuntos de *transformações quânticas universais* \cite{chuang}.

Além destas, CCs podem conter outras operações como, por exemplo, operações de medida (neste caso, interpretadas por projeções do vetor correspondente ao q-bit sobre um par de subespaços ortogonais) e operações de controle.

A TQ elementar Hadamard (H) quando aplicada no estado clássico $|0\rangle$, gera a superposição $|0\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, cuja expressão matricial é dada por

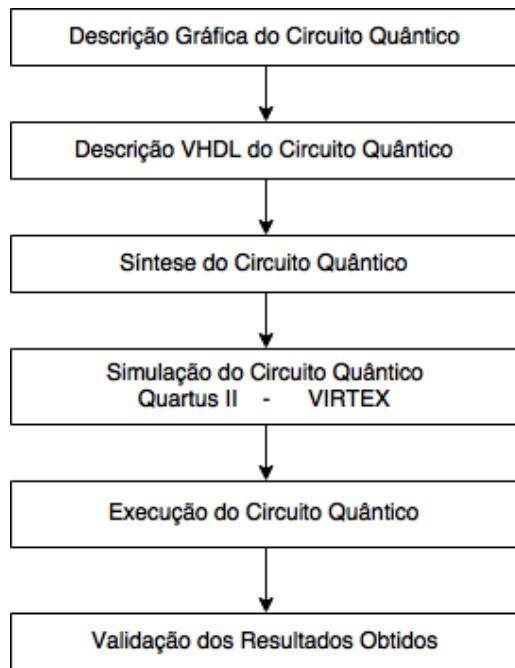
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

A representação de portas lógicas quânticas atuando sobre n q-bits envolve matrizes de dimensão $2^n \times 2^n$ obtido pelo produto tensor. Assim, a aplicação simultânea de uma porta H^2 sobre um q-bit phi + psi é indicada por $H^{\otimes 2}(|\phi\rangle \otimes |\psi\rangle) = H|\phi\rangle \otimes H|\psi\rangle$.

Além destas operações, tem-se classe de portas lógicas que atuam sobre dois ou mais q-bits sem serem fatoradas como produtos tensorial. Casos particulares, de TQs atuantes em dois q-bit são as portas *controladas* e os *swaps* (Nielsen; Chuang, 2000). CNOT, como também é chamado o não-controlado, recebe um q-bit de controle e um q-bit alvo: se o q-bit de controle receber $|1\rangle$, o estado do q-bit alvo é trocado, senão o q-bit alvo permanece inalterado. Genericamente, se U é um operador unitário de um q-bit, uma operação *U-controlada* é uma operação a dois q-bits, sendo um q-bit de controle e o outro o q-bit alvo recebendo a ação do operador U .

-Modelagem de Circuitos Quânticos Utilizando VHDL:

A técnica em PLDs (*Programmable Logic Device*), provê uma metodologia para projeto e descrição de circuitos digitais em HDL, as quais estão baseadas na estrutura e no comportamento de seus componentes, manipulando equações booleanas, tabelas-verdade e operações complexas de maneira direta. Considerando esta técnica de projeto, as etapas de desenvolvimento do projeto qExVHDL estão resumidas na figura abaixo.



Pela geração do código VHDL e aplicação de FPGAs, tem-se uma opção para obter um incremento na velocidade e no número de q-bits quando da simulação quântica, fornecendo uma alternativa de análise e desenvolvimento de algoritmos quânticos.

Para geração do código VHDL~\cite{altera,vhdl:tutorial,husemann,pedroni}, utilizou-se a ferramenta Quartus II, desenvolvida pela empresa Altera (ZHANG, 2001), permitindo que sejam projetado e simulado em um FPGA (EP4CGX150DF31C7N) capaz de validar o código desenvolvido para simulação dos circuitos estudados.

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 evitar a limitação gerada pelo acesso à memória sequencial, torna-se uma linguagem atrativa para execução de algoritmos paralelos. Esta característica é de fundamental importância ao paradigma quântico.

A extensão da linguagem VHDL para aplicação na descrição do comportamento e a estrutura de CQs inclui também o desenvolvimento de bibliotecas de portas quânticas em VHDL. A modelagem contempla a construção, métodos, dados e componentes. Como estas necessidades não são supridas pelas ferramentas utilizadas, foi necessário a criação de uma série de bibliotecas. A correção do circuito pode ser verificada ou por simulação via *software* ou por simulação em FPGA. A modelagem de circuitos quânticos é descrita em VHDL e então sintetizada em *hardware* como processo de validação. Aplicam-se tecnologias clássicas na construção de componentes fundamentais dos CQs e nas regras que governam as computações quânticas.

3. RESULTADOS E DISCUSSÃO

Na implementação das bibliotecas criadas neste trabalho, utilizou-se a ferramenta Quartus II desenvolvida pela empresa Altera. A partir da descrição

de circuitos em VHDL, pode-se estabelecer o fluxo de projeto para a prototipação em FPGA ou ASIC (*Application-Specific Integrated Circuit*).

Neste trabalho, utiliza-se um FPGA (*EP4CGX150DF31C7N*, *Família: Cyclone IV*) para a prototipação de CQs, cujas bibliotecas agregam as seguintes especificações: (i) dados do tipo *complex32* de 64 bits, apresentando as partes reais e imaginárias de cada número, e para tal utiliza-se um número de ponto flutuante de 32 bits para cada uma delas; (ii) funções matemáticas específicas providas pelas bibliotecas de ponto flutuante, números complexos e matrizes desenvolvidas durante a realização deste trabalho;

O tipo *complex32*, provê descrição para as amplitudes associadas a um q-bit definida por um par de números complexos normalizados (c0,c1), e cada complexo com suas correspondentes partes real e imaginária ($c_0, c_1 = ((c_0.r, c_0.i), (c_1.r, c_1.i))$). Sistemas unidimensionais, cujos estados são definidos por um q-bit, estão descritos pelo tipo *complex32vector*. E, sistemas bidimensionais são descritos pelo tipo *complex32matrix*.

4. CONCLUSÕES

O gerenciamento de recursos na simulação quântica via *hardware* torna-se uma estratégia importante, pois integra conceitos e tecnologias da física quântica com demanda exponencial de recursos quando da simulação via tecnologias clássicas.

Neste contexto, o qExVHDL vem colaborar com a discussão dos conceitos, tecnologias e aplicações da simulação quântica via *hardware* enquanto área estratégica para o desenvolvimento científico e tecnológico. Busca-se difundir a metodologia da programação quântica em sistemas digitais pelo estudo de casos, aplicados a algoritmos básicos e CQs.

Na continuidade, tem-se a implementação do processo probabilístico da operação de medida, incluindo a otimização da implementação e análise de erros. A análise da metodologia da programação quântica em sistemas digitais e o processamento de sinais é outro grande desafio, com aplicações em sistemas de comunicação.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- NIELSEN, M. A. AND CHUANG, I. L. **Quantum Computation and Quantum Information**. Cambridge, Inglaterra: Cambridge University Press, 2000.
- BISHOP, D. **Floating point package user's guide**. New Jersey, Estados Unidos da América: Packages and bodies for the IEEE, 2008.
- ZHANG, W. **VHDL tutorial: Learn by example. v.1. p.420**. Riverside, Estados Unidos: University of California, 2001.
- CONCEIÇÃO, C. M. O. **Uma arquitetura de co-processador para simulação de algoritmos quânticos em FPGA**. Porto Alegre, Brasil: Editora da UFRGS, 2013.
- MONTEIRO, E. et al. qExVHDL: Uma Simulação Quântica em VHDL. In: **XXXV CLEI**, 35., Pelotas, Brasil. Anais... Editora da UFPEL, Pelotas, 2009. v.1. p.1-8.