

qExVHDL: Uma Simulação Quântica em VHDL

Eduarda Monteiro, Diego Jaccottet, Thiago Pontes *
Renata Reiser, Eduardo Costa, Maurício Pilla

¹Centro Politécnico – Universidade Católica de Pelotas (UCPel)
Rua Felix da Cunha, 412 – Pelotas – RS – Brazil

{eduardam, diegopj, reiser, ecosta}@ucpel.tche.br
thiago0b12@gmail.com, pilla@ufpel.edu.br

Abstract. *This paper applies the qExVHDL methodology integrated with a quantum state register architecture in order to perform simulation of basic quantum algorithms considering established standards of classic circuits.*

Resumo. *Este trabalho considera a aplicação da metodologia qExVHDL integrada a uma arquitetura de registrador de estado quântico para viabilizar a simulação de algoritmos básicos da computação quântica, considerando os padrões estabelecidos para os circuitos clássicos.*

1. Introdução

A descrição de circuitos quânticos a partir dos padrões estabelecidos para os circuitos clássicos tem contribuído para compreensão do paralelismo quântico familiar ao paradigma da programação quântica. Neste sentido, considera-se relevante o estudo de arquiteturas paralelas para a simulação via hardware [Khalid et al. 2004, Gothandaraman et al. 2008, Aminian et al. 2008] de algoritmos básicos da computação quântica [Nielsen and Chuang 2000], viabilizando o desenvolvimento de novos algoritmos, independente do avanço tecnológico. O principal objetivo deste trabalho é analisar os resultados que se obtiveram com a modelagem e a implementação de métodos para manipulação de dados e controle quânticos. A implementação está baseada na metodologia denominada qExVHDL (*Quantum Extension for VHDL*) obtendo a descrição de circuitos quânticos em VHDL e sua correspondente execução em FPGA (*Field Programmable Gate Array*). A linguagem VHDL (*VHSIC Hardware Description Language*), consiste em um padrão estabelecido pela IEEE para realização de simulações, sínteses e propriedade intelectual, sendo portátil e permitindo a reutilização de componentes em projetos de *hardware*. Para validação da metodologia qExVHDL, vem sendo desenvolvidos estudos de caso que simulam algoritmos quânticos básicos, os quais estão baseados em propriedades como interferência, paralelismo e superposição de estados. Assim, com suas correspondentes sínteses em FPGA, a versão quântica do Interferômetro Mach-Zehnder (IMZ), algoritmo de Deutsch (AD) e o algoritmo de Grover (AG) são considerados. O trabalho também apresenta um estudo dos erros computacionais que ocorreram durante as simulações, incluindo uma análise das áreas dos circuitos correspondendo ao número de elementos lógicos gerados, frequência máxima de operação e trabalhos relacionados.

O artigo continua com a descrição da modelagem da computação quântica em VHDL na seção 2, incluindo a simulação de bits quânticos e o processo de geração automática do código em VHDL. Por fim, os resultados alcançados na implementação da

*Bolsista PIBIC/CNPq

metodologia qExVHDL e uma comparação de recursos utilizados por trabalhos relacionados são descritos na seção 2.3.

2. Modelando Computação Quântica em VHDL

Para a aplicação da linguagem VHDL, considerou-se o desenvolvimento de operadores constituídos: (i) de portas quânticas, as quais foram construídas a partir de métodos, dados e componentes providos por esta linguagem; e (ii) de regras, governando a sincronização e a evolução temporal das simulações de computações quânticas.

2.1. Simulação de bits quânticos

Na especificação de um bit quântico $|\psi\rangle = a|0\rangle + b|1\rangle$, consideram-se pares de números reais: $|\psi\rangle = (\alpha_1, \alpha_2)|0\rangle + (\beta_1, \beta_2)|1\rangle$, onde α_1 e α_2 correspondem à parte real e imaginária do complexo a , coeficiente da primeira componente da base, e β_1 e β_2 correspondem à parte real e imaginária do complexo b , coeficiente da segunda componente. Esta representação introduz erro nos dados de entrada do sistema modelado. Na implementação dos valores reais $\alpha_1, \alpha_2, \beta_1, \beta_2$ foram utilizados dois sistemas de ponto flutuante: (i) 32 bits - 1 bit de sinal, 8 bits de expoente e 23 bits de mantissa (IMZ e AD) e (ii) 8 bits - 1 bit de sinal, 4 bits de expoente e 3 bits de mantissa (AG). Os coeficientes a e b foram obtidos considerando registradores de estados quânticos com quatro partes, pela arquitetura na Fig. 1.



Figura 1. Arquitetura de Registrador de Estado (1 q-bit)

2.2. Geração Automática de Código VHDL

A metodologia qExVHDL considera a geração automática de descrições em VHDL, a nível comportamental. Esta característica foi consolidada a partir da customização da ferramenta RAMAGT (*Radix-2^m Array Multipliers Automatic Generation Tool*), proposta em [Porto et al. 2008], desenvolvida pela integração de duas linguagens: (i) a linguagem C++, responsável pela construção da interface gráfica; e (ii) a linguagem CSL (*C Scripting Language*), responsável pela descrição em VHDL.

Na Fig. 2, tem-se a arquitetura da ferramenta *Quantum-RAMAGT* incluindo as interfaces para geração de código VHDL, correspondendo a quatro etapas caracterizadas pela suas funcionalidades: (i) interface principal do programa; (ii) a seleção das otimizações disponíveis; (iii) a seleção dos parâmetros que definem a precisão dos dados, a dimensão do sistema e o número de operadores para a aplicação corrente; (iv) finalização do código VHDL e geração do arquivo descritor do projeto.

A descrição dos circuitos quânticos é obtida pela aplicação de componentes (multiplicadores e somadores) de ponto flutuante disponibilizados pela ferramenta *ISE* da empresa *Xilinx*. Para viabilizar a síntese do circuito implementado, as seguintes otimizações foram desenvolvidas:



Figura 2. Arquitetura e Interfaces da *Quantum-RAMAGT*.

1. Simplificação dos coeficientes, no tratamento de números complexos, e modelagem de acumulador para armazenamento de somas, consistindo na atribuição, de operações de adição a uma única variável;
2. Tratamento de operações envolvendo multiplicações: (i) no caso da constante 0, foram desconsideradas; e (ii) no caso da constante 1, considerou-se o número de seu multiplicador (multiplicando) como produto;
3. Tratamento de operações envolvendo multiplicações pela constante -1 , obtida pela negação do bit de sinal;
4. Aplicação de conversores da precisão dos dados permitindo que a precisão dos parâmetros de entradas e saídas, possa ser distinta e definida pelo usuário;
5. Omissão dos componentes conversores da precisão dos dados e tratamento dos parâmetros de entrada que envolvem multiplicações pela constante 0.

2.3. Resultados e Trabalhos Relacionados

A metodologia $\mathcal{QExVHDL}$ foi aplicada na descrição do Interferômetro de Mach-Zehnder (IMZ) quer [Monteiro et al. 2008], algoritmo de Deutsch [Monteiro et al. 2009a] e o algoritmo de Grover [Monteiro et al. 2009b], respectivamente. A validação das descrições em VHDL dos circuitos foi sintetizada em ambiente *Xilinx* na placa FPGA: *XC2V P30*, Família *Virtex-II PRO*, Pacote *FF896*, Área total correspondente a 30816 elementos lógicos. Esta seção apresenta os principais resultados comparando estes três estudos de caso e trabalhos relacionados. De acordo com a Tabela 1, todos os estudos de caso foram sintetizados em ambiente *Xilinx*, considerando como parâmetros a precisão numérica, a frequência máxima de operação de cada um dos circuitos, o número de q-bits manipulados, percentual de utilização de área, e a utilização de gerador de código.

As otimizações aplicadas justificam as seguintes observações: verifica-se uma semelhança, na frequência máxima de operação e no percentual de utilização de área, entre os algoritmos do IMZ (1 q-bit) e AD (2 q-bits). No AG (3 q-bits) a implementação apresenta considerável diminuição do percentual de utilização de área considerando que se trata de um sistema tridimensional e com maior número de operadores. Por requerer um número maior de operações, este circuito se torna vulnerável a um maior caminho crítico, ocorrendo um aumento na frequência máxima de operação e uma diminuição no percentual de utilização de área. Na Tabela 2, verifica-se que os trabalhos relacionados, citados na introdução, dispõem de melhores recursos computacionais, trabalham com ponto fixo, reduzindo o custo computacional mas com menor precisão nos resultados numéricos, e podem ou não utilizar ferramentas de geração automática de código.

Em continuidade, estuda-se a aplicação da metodologia aos algoritmos de Shor e Teleportação Quântica. As novas otimizações na $\mathcal{QExVHDL}$ serão realizadas no nível

Tabela 1. Comparação de Simulações

Caso	Q-bits	Mantissa	Gerador	Área	Frequência	Percentual
IMZ	1	23 bits	Sim	12425	151,871MHz	40,3%
Deutsch	2	23 bits	Sim	12555	151,871MHz	40,7%
Grover	3	3 bits	Sim	4859	151,838MHz	16,0%

Tabela 2. Comparação de Recursos - Simulação Quântica via Hardware

Trabalho	Linguagem	Gerador	Mantissa	Placa	El. Lógicos
1 (Khalid)	VHDL	Sim	Ponto Fixo 16 bits	Altera Stratix EP1S80B956C6	79040
2 (Gothandaraman)	VHDL	Não	Ponto Fixo 52 - 64 bits	Virtex-4 LX160 Virtex-II PRO XC2VP30	152064 30816
3 (Aminian)	VHDL	Não	Ponto Fixo 8 - 16 bits	Altera Stratix EP1S80B956C6	79040

RT (Transferência de Registradores), cujo projeto consiste na interligação dos blocos para a caracterização dos circuitos dedicados. A análise da metodologia da programação quântica em sistemas digitais, incluindo o processamento de sinais, é outro desafio em sistemas de comunicação, com aplicações específicas de circuitos integrados na prototipação em ASIC (*Application-Specific Integrated Circuit*).

Referências

- Aminian, M., Saeedi, M., Zamani, M. S., and Sedighi, M. (2008). FPGA-based circuit model emulation of quantum algorithms. In [Aminian et al. 2008], pages 399–404.
- Gothandaraman, A., Peterson, G. D., Warren, G. L., Hinde, R. J., and Harrison, R. J. (2008). Fpga acceleration of a quantum monte carlo application. *Parallel Comput.*, 34(4-5):278–291.
- Khalid, A. U., Zilic, Z., and Radecka, K. (2004). Fpga emulation of quantum circuits. In *ICCD '04: Proceedings of the IEEE International Conference on Computer Design*, pages 310–315, Washington, DC, USA. IEEE Computer Society.
- Monteiro, E. R., Jaccottet, D. P., Reiser, R. H. S., da Costa, E. A. C., and Pilla, M. L. (2009a). Simulação quântica em vhdL: Um estudo de caso baseado no algoritmo de deutsch. In *CLEI 2009 - XXXV Conferência Latino-Americana de Informática*, Pelotas, Brasil.
- Monteiro, E. R., Jaccottet, D. P., Reiser, R. H. S., da Costa, E. A. C., and Pilla, M. L. (2009b). Simulação quântica em vhdL: Um estudo de caso baseado no algoritmo de grover. In *WSCAD-SSC: X Simpósio em Sistemas Computacionais*, São Paulo, Brasil.
- Monteiro, E. R., Jaccottet, D. P., Reiser, R. H. S., da Rocha Costa, A. C., da Costa, E. A. C., and Pilla, M. L. (2008). Aplicação da biblioteca qexvhdL na descrição do interferômetro de mach-zehnder. In *Jornadas Chilenas de Computación 2008*, Punta Arenas, Chile.
- Nielsen, M. A. and Chuang, I. L. (2000). *Quantum Computation and Quantum Information*. Cambridge University Press.
- Porto, D., Roschild, J., COSTA, E. A. C., and ALMEIDA, S. J. M. (2008). Ramagt-radix-2^m - array multipliers automatic generation tool. In *XXIII South Symp. on Microelectronics*, pages 65–68.