INTERFACE GRAFICA PARA REDES DE TRANSISTORES MAPEADAS PARA UM GRAFO

DIOGO DALPIAN DOS SANTOS[1]; LEOMAR SOARES DA ROSA JUNIOR[2]; FELIPE DE SOUZA MARQUES[3]

[1]Universidade Federal de Pelotas – diogo_d.santos@hotmail.com
[2]Universidade Federal de Pelotas – leomarjr@inf.ufpel.edu.br
[3]Universidade Federal de Pelotas – felipem@inf.ufpel.edu.br

1. INTRODUÇÃO

Circuitos digitais estão cada vez mais presentes no ambiente profissional e pessoal, prova disso é a renda de R$ 1 trilhão ao ano do mercado de eletrônicos em todo mundo. Praticamente todas as áreas da nossa vida se beneficiam dessa tecnologia, como por exemplo: medicina, automação, robótica, entretenimento, transportes, computadores, televisores, telefones celulares e assim por diante. Toda essa abrangência tecnológica se dá ao fato de os preços e tamanhos desses circuitos terem diminuído consideravelmente.

Processador	Ano	Tamanho (nanômetros)
Pentium 4	*2004*	*180 nm*
Pentium D	*2005*	*90 nm*
Core 2 Duo	*2007*	*65 nm*
Core I3, I5, I7 (1ª geração)	*2008*	*45 nm*
Core I3, I5, I7 (2ª geração)	*2011*	*32 nm*
Core I3, I5, I7 (3ª geração)	*2012*	*22 nm*
Core I3, I5, I7 (4ª geração)	*2013*	*22 nm*

Tabela 1. Tamanho dos processadores Intel ao longo dos anos (2004-2013)

A miniaturização dos circuitos e o avanço na tecnologia de fabricação tornou sua concepção complexa, fazendo com que seja indispensável o uso de ferramentas que facilitem o projeto de um novo circuito. Esse componente eletrônico é composto por redes de transistores que utilizam a matemática binária para representar uma certa função lógica.

Uma rede de transistores pode ser mapeada para um grafo (estrutura de dados que representam objetos, onde cada aresta desse grafo corresponde a um transistor). Dessa forma o objetivo desse trabalho é apresentar uma proposta de interface para essa rede que está sendo simulada através de um grafo. Para isso será utilizado um algoritmo de compactação de arestas (POSSANI, 2011), o qual reconhece arranjos de transistores do tipo série-paralelo e também os conhecidos como redes *Wheatstone Bridge* – arranjo em que não se pode afirmar se a relação entre os transistores é do tipo série ou paralelo.

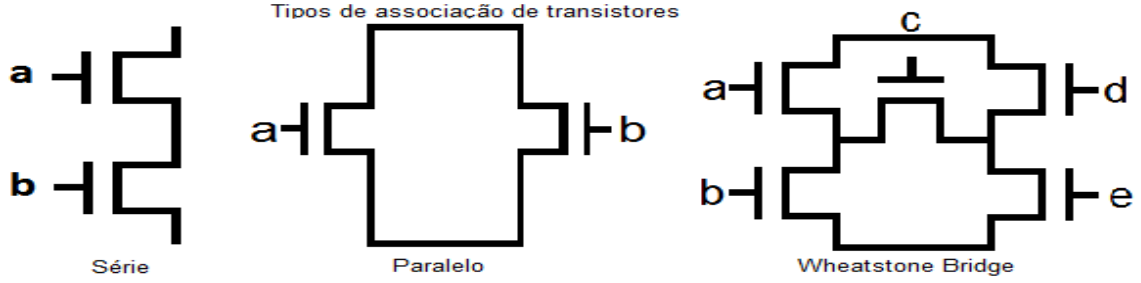

Figura 1. Tipos de relação entre os transistores.

Também é desejável que essa interface possibilite realizar modificações no grafo, podendo, dessa forma, personalizar de acordo com os interesses do projeto e que facilite o entendimento do usuário criando um visual simples e prático.

Ao final do desenvolvimento deseja-se integrar a ferramenta ao ambiente SwitchCraft (CALLEGARO, 2009) – segundo ele, "o ambiente SwitchCraft provê um conjunto de ferramentas para geração de redes de chaves lógicas".

2. METODOLOGIA

Para o desenvolvimento desse trabalho a linguagem de programação escolhida foi Java, visando uma possibilidade de integração com o ambiente SwitchCraft que é escrito inteiramente com a mesma. Visto que essa linguagem é orientada a objetos, também foi necessário o reforço com os conceitos desse método de programação. Também foi necessário definir a *framework* a ser utilizada para a criação da interface gráfica, e ficou decidido pela *Java Universal Network Graph* (JUNG), a qual fornece ferramentas para representação e manipulação de grafos de forma eficiente e flexível.

A partir desse momento pode-se iniciar a criação de um algoritmo para a representação gráfica de uma rede de transistores. Para isso é preciso traduzir um grafo, representado em uma estrutura genérica de matriz de adjacência, para a estrutura JUNG, a fim de facilitar a manipulação em cima dos vértices e arestas.

A próxima etapa consiste em utilizar um algoritmo de compactação de arestas (POSSANI, 2011) para que se possa identificá-las (transistores) como sendo série ou paralelo. A figura 2 a seguir mostra o funcionamento do mesmo.

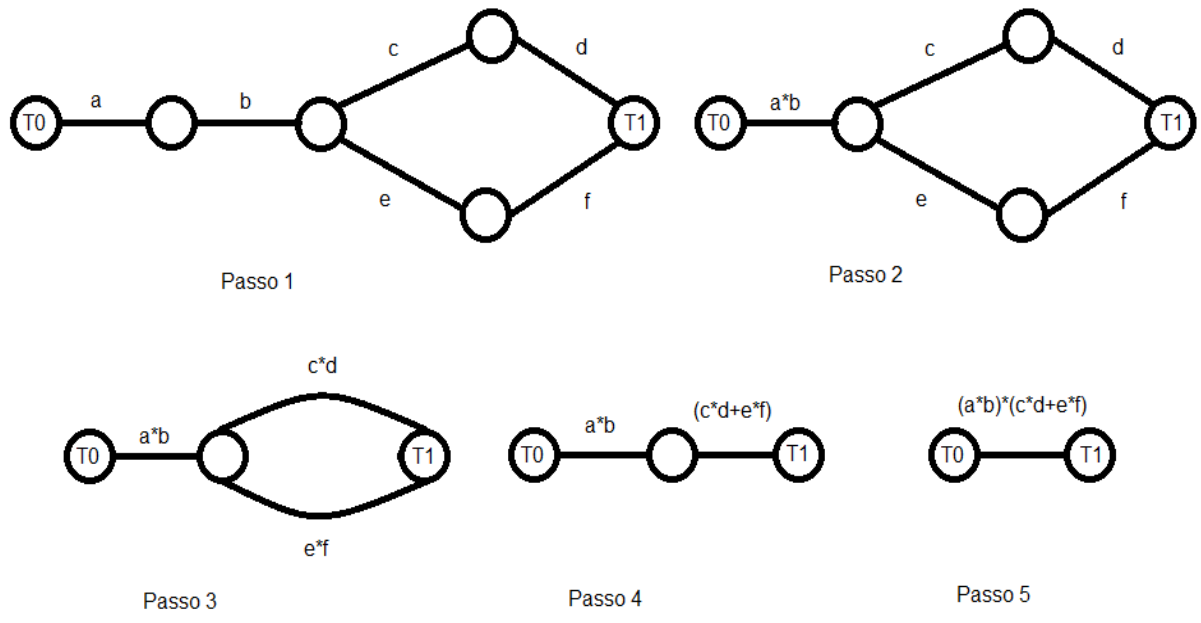

Figura 2. Passos para a aplicação do algoritmo de compactação de arestas.

Como pode ser visto na figura 2, o algoritmo percorre o grafo diversas vezes, partindo do vértice terminal *T0* até o vértice terminal *T1*, buscando a primeira vez por arestas em série (Passos 1, 2 e 3), quando não existirem mais arestas em série o grafo é percorrido uma segunda vez identificando arestas em paralelo (Passo 4), da mesma forma, quando não existirem mais arestas em paralelo o grafo é percorrido pela terceira vez, buscando novamente por arestas em série (Passo 5).

O algoritmo para quando não é mais possível compactar as arestas em série e paralelo, ou seja, o tamanho do grafo antes da compactação permanece imutável após a aplicação do algoritmo.

Por fim, estando o grafo totalmente compactado e as arestas identificadas como série e paralelo, é possível posicionar vértices e arestas dentro de uma janela gráfica e visualizar a rede de transistores, realizando o processo inverso ao do algoritmo de compactação.

3. RESULTADOS E DISCUSSÃO

Até a escrita desse trabalho foi implementado o algoritmo que faz a tradução do grafo de uma estrutura de matriz de adjacência para a estrutura JUNG, assim como o algoritmo que compacta esse grafo traduzido.

Com a continuação do trabalho pretende-se finalizar a parte de posicionamento dos vértices e arestas a fim de adicionar ferramentas que permitam a personalização em cima da rede de transistores que já está representada graficamente.

O objetivo final é, se possível, integrar o visualizador implementado a ferramenta *SwitchCraft* (CALLEGARO, 2009).

4. CONCLUSÕES

O principal desafio no momento é realizar o posicionamento dos vértices e arestas dentro da janela gráfica.

Ao término do desenvolvimento desse projeto pretende-se chegar a um visualizador melhor que o apresentado pela ferramenta para o qual está sendo desenvolvido e, posteriormente integrá-lo a mesma.

Dessa forma será possível auxiliar os projetistas de hardware na concepção de circuitos digitais e, consequentemente, contribuindo para o desenvolvimento tecnológico em todos os setores que afetam nossas vidas.

5. REFERÊNCIAS BIBLIOGRÁFICAS

GOODRICH, T. M.; TAMASSIA, R. **Data Structures and Algorithms in Java**. Wiley, 2005. 4 v.

SANTOS, R.; GRÉGIO, A. R. A. Introdução à representação e análise de grafos com a API JUNG. **Revista MundoJ**, v.49, p. 35 - 45, 01 set. 2011.

JUNIOR, D. S. J. Geração de redes de transistores otimizadas utilizando uma abordagem baseada em grafos. **Revista Eletrônica de Iniciação Cietífica**, v. 11, n. 4, 2011.

VINICIUS, C. **SwitchCraft: Um ambiente computacional para síntese e análise de redes lógicas**. 2009. Trabalho de conclusão de curso (Graduado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul.

MARIN, S. E. **Fluzz – redes sociais: Geração, visualização e buscas que maximizam a probabilidade de influência entre indivíduos**; 2013; Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Goiás.