

MINIMIZAÇÃO LÓGICA: A IMPLEMENTAÇÃO DE UM MÉTODO TABULAR

¹ADRIANO TOLFO DOTTA; ²GUSTAVO MACEDO DOS SANTOS; ³REGINALDO DA NÓBREGA TAVARES

¹Universidade Federal de Pelotas – adrianot.dotta@gmail.com

²Universidade Federal de Pelotas – gustavo.santos120201@gmail.com

³Universidade Federal de Pelotas – regi.ntavares@gmail.com

1. INTRODUÇÃO

Este trabalho descreve o projeto de um programa de computador que será utilizado como ferramenta de auxílio da disciplina de Eletrônica Digital. A disciplina de Eletrônica Digital é componente curricular dos cursos de Engenharia Eletrônica e de Engenharia de Controle e Automação da UFPel. O grupo de trabalho é formado por estudantes e o professor da disciplina de Eletrônica Digital.

O programa de computador está escrito em linguagem de programação C (HERBERT, 1997). Esta linguagem é amplamente utilizada em projetos de engenharia e de computação. O objetivo do programa é implementar o método de minimização lógica conhecido como Quine-McCluskey (McCluskey, 1986), método apresentado na disciplina de Eletrônica Digital. A minimização lógica é uma etapa fundamental do projeto de circuitos lógicos, e, por esta razão, é conteúdo importante na formação do engenheiro projetista de circuitos lógicos digitais. Depois de pronto, o programa de computador deverá ser utilizado em sala de aula. Portanto, este programa de computador tem objetivos educacionais.

O método Quine-McCluskey utiliza tabelas de verificação para armazenar termos lógicos. Os termos lógicos são armazenados e comparados entre si. Porém, as comparações devem ser realizadas levando em conta algumas regras de operação lógica. O objetivo das comparações é eliminar as variáveis lógicas redundantes que podem estar contidas em um par de termos.

Apesar de útil, este método é exaustivamente repetitivo, e, por esta razão, sujeito a erros quando o estudante de Eletrônica Digital está aplicando o método.

Para implementar as tabelas de verificação utilizamos cadeias de bits a fim de aproveitar melhor a memória do computador. Portanto, a investigação em andamento procura por maneiras de implementar as tabelas de verificação. O objetivo é implementar as tabelas com alguma eficiência do consumo de memória. Porém, este trabalho está em andamento.

2. METODOLOGIA

Foram realizados encontros semanais do grupo desde maio de 2019. Nestes encontros foram discutidas formas de implementar o programa e torná-lo funcional.

No período de pesquisa e aprendizado do projeto, nós estudantes estávamos cursando a disciplina de Programação de Computadores e Eletrônica Digital. Isto foi de grande valia, pois as experiências adquiridas no projeto ajudaram a expandir os conhecimentos sobre a programação de computadores, e

relacionar aprendizados nos encontros do projeto com os encontros das aulas da disciplina de Programação de Computadores e Eletrônica Digital.

3. RESULTADOS E DISCUSSÃO

O método de minimização lógica Quine-McCluskey utiliza tabelas de verificação onde cada linha da tabela armazena um termo lógico. Por sua vez, um termo lógico é composto por um produto de variáveis lógicas, como o termo $(x.y.z)$, onde x , y e z são representações de diferentes variáveis lógicas. Cada variável lógica de um termo pode assumir um valor entre dois valores lógicos: “um” e “zero”. Os termos lógicos são agrupados na tabela de acordo com a representação de suas variáveis. Neste programa assumimos que na sua forma direta a variável $x=1$ assume a expressão x , e na sua forma complementar a variável assume a expressão x' na qual, $x=0$ e $x'=1$. Assim, as tabelas serão formadas por diferentes grupos de termos: termos que possuem apenas uma variável com valor um, termos com duas variáveis com valor um, termos com três variáveis com valor um, etc.

As comparações são realizadas entre duas linhas da tabela de verificação, quer dizer, com dois termos entre vários termos armazenados na tabela. O resultado de cada comparação é armazenado em uma nova tabela de verificação. Assim, o programa executa o método com duas tabelas ativas: uma tabela de comparação e outra tabela para armazenar resultados. Os termos já comparados e que não podem mais ser reduzidos serão marcados e utilizados na segunda etapa do método. Estes termos são armazenados em um vetor próprio para utilização na parte final do método. Após a realização de todas as possíveis comparações, a tabela que contém os novos resultados passa a ser a tabela de comparações, e a tabela de comparações anterior passa a armazenar os novos resultados.

Desta forma, a investigação em andamento procura por modos de implementar as tabelas de verificação. Na atual implementação das tabelas de verificação, utilizamos cadeias de *bits* para representar os termos ao invés de representar o valor lógico zero e o valor lógico um por números inteiros. Cabe chamar a atenção que na linguagem de programação C, um valor inteiro pode ser implementado por vários *bytes*¹, e, por esta razão consumir vários endereços de memória. O objetivo da estratégia deste programa é reduzir o consumo de posições de memória. As tabelas são alocadas com 2^n , onde n é o número de variáveis lógicas representadas. Assim, neste projeto estamos investigando maneiras de aproveitar melhor a memória do computador, muito embora, os computadores pessoais (de mesa e *notebooks*) dispõem de grande quantidade de memória. No entanto, a memória do computador é compartilhada por um grande número de programas que podem estar em execução.

O programa mostra passo-a-passo o método: os termos agrupados, as tabelas onde os termos ficarão armazenados para comparação. Como o desenvolvimento do programa ainda está em andamento, o mapa final do método ainda não está implementado. Consequentemente, ainda não é mostrado para o usuário o resultado final.

¹ Um *byte* corresponde a uma cadeia de 8 bits, e um dígito binário é chamado de *bit*.

O programa mostra os passos da minimização lógica para o estudante, desta maneira o estudante poderá visualizar o que está sendo acontecendo, podendo ele mesmo conferir as comparações realizadas pelo programa, corrigir exercícios que ele mesmo fez a mão, tirar alguma dúvida referente a alguma operação realizada.

Por fim, o programa oferece um menu de interação, onde o usuário pode escolher a operação que deseja realizar. Neste momento o programa implementa as opções: inserir termos, inserir termos a partir de um arquivo, executar novamente o processo de minimização e encerrar o programa. Na próxima versão do programa outras opções deverão ser colocadas no menu.

4. CONCLUSÕES

O projeto propõem uma ferramenta de computador para implementar o método de minimização lógica de Quine-McCluskey. A ferramenta mostra para o estudante o passo-a-passo das etapas de minimização, as tabelas de verificação geradas, e as comparações entre termos. Acreditamos que a ferramenta pode ajudar a exposição de conteúdos em sala de aula e incentivar os estudantes a aprender um método de minimização lógica.

Para que possa funcionar com melhor eficiência computacional, estamos trabalhando em melhorias no código em linguagem de programação C, neste sentido, o objetivo é reduzir o consumo da memória disponível do computador. Ainda falta completar a parte final do método.

Esta é a primeira versão do programa e pretendemos evoluir a ferramenta.

5. REFERÊNCIAS BIBLIOGRÁFICAS

HERBERT, S. **C Completo e Total**. São Paulo: MAKRON Books Ltda., 1997.

McCLUSKEY, E. **Logic design principles**. Englewood, New Jersey: Prentice-Hall, 1986.