

OBTENÇÃO DE IMPLICANTES PRIMOS PARA FUNÇÕES BOOLEANAS ATRAVÉS DA OPERAÇÃO DE CONSENSO

Alexandre César Rodrigues da Silva¹, Ivanil Sebastião Bonatti² e Cláudio Kitano³

Resumo — No desenvolvimento de projetos de sistemas digitais envolvendo FPGA, necessita-se de eficientes algoritmos computacionais para a minimização das funções booleanas contendo um grande número de variáveis. Neste trabalho apresenta-se o algoritmo GeraPlex, que obtém todos os implicantes primos de uma função booleana através de sucessivas aplicações da operação de consenso aos ramos de uma árvore de decisão binária, onde cada caminho (nó até folha) representa um mintermo ou irrelevante da função. Da aplicação do consenso são definidas as operações de fusão, deslocamento e expansão, que diminui, mantém e aumenta, respectivamente, os caminhos da árvore. No final do processo os caminhos que permanecem na árvore representam os implicantes primos da função. Dezenas de funções foram minimizadas e os resultados obtidos comparados com o tradicional método tabular para a geração de implicantes primos. Em todos os casos o GeraPlex obteve os mesmos implicantes primos e apresentou desempenho superior em tempo e memória.

Palavras Chave — Consenso iterativo, Minimização de funções Booleanas, Otimização, Síntese lógica.

INTRODUÇÃO

Na simplificação de funções booleanas de médio porte (tipicamente de 10 a 30 variáveis) o número de implicantes primos gerados pode crescer exponencialmente. Portanto a obtenção de todos os implicantes primos requer grande quantidade de memória e tempo de execução. Alguns autores trataram o problema de forma algébrica [1], [2], enquanto outros utilizaram-se de tabelas e árvores de decisão binária [3], [4], [5] e [6].

Quine [6] e Tison [7] utilizaram-se do consenso iterativo para obter todos os implicantes primos de uma dada função booleana. Neste trabalho, apresenta-se o algoritmo denominado GeraPlex, que obtém todos os implicantes primos de uma função booleana através de sucessivas aplicações da operação de consenso aos ramos de uma árvore de decisão binária que representa a tabela verdade da função a ser minimizada. Nesta árvore cada caminho, que vai da raiz à folha, representa um mintermo ou irrelevante da função.

No algoritmo proposto da aplicação do consenso aos ramos da árvore são definidas três operações: fusão,

expansão e deslocamento que diminui, aumenta e mantém o número de caminhos da árvore, respectivamente.

Resultados experimentais indicam que o método proposto é uma eficiente alternativa para a resolução do problema considerado, pois é de fácil implementação e muito adequado para representar circuitos lógicos combinacionais.

O CONSENSO DE UMA FUNÇÃO

O Consenso de $f(x_1, x_2, \dots, x_i, \dots, x_n)$ com relação à variável x_i é definida como $C_{x_i}(f) = f_{x_i} \cdot f_{\bar{x}_i}$ [9]. Dessa forma, pode-se afirmar que o consenso de uma função booleana com relação a uma variável representa o componente que é independente dessa variável. O consenso pode ser estendido para conjuntos de variáveis como uma aplicação iterativa da operação de consenso sobre as variáveis do conjunto. Por exemplo, o consenso entre os termos produtos $A \cdot B \cdot D \cdot E$ e $A \cdot B \cdot C \cdot D$ é o termo produto $B \cdot C \cdot D \cdot E$. Não existe o consenso entre os termos produtos $A \cdot B$ e $A \cdot B \cdot C$, pois deve haver uma única variável que ocorre negada em um dos termos e não negada no outro.

Método do Consenso para Obter Implicantes Primos

O método do consenso iterativo [10] para a obtenção dos implicantes primos de uma função booleana consiste na aplicação das operações de eliminação e consenso, até que estas operações não sejam mais aplicáveis. Na operação de eliminação elimina-se quaisquer termos produto que cobre algum outro termo produto e na operação de consenso acrescenta-se como termos produto o consenso de dois termos produtos, se este não for coberto por nenhum outro termo produto contido na soma de produto.

Como exemplo da aplicação do método do consenso considere a função $F_1(A, B, C, D) = \sum m(5, 7, 8, 9, 13)$. Esta função $F_1(A, B, C, D)$ pode ser representada através da soma de produto como $A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot \bar{D} + A \cdot B \cdot \bar{C} \cdot D + A \cdot B \cdot \bar{C} \cdot \bar{D} + A \cdot B \cdot C \cdot D$. O consenso entre $A \cdot B \cdot C \cdot D$ e $A \cdot B \cdot \bar{C} \cdot D$ resulta no termo $A \cdot B \cdot C$ que não é coberto por nenhum dos termos produtos, devendo então ser incluído na soma. Com a inclusão do termo consenso tem-se uma nova expressão soma de produto, ou seja, $F_1(A, B, C, D) = A \cdot B \cdot C \cdot D + A \cdot B \cdot C \cdot \bar{D} + A \cdot B \cdot \bar{C} \cdot D + A \cdot B \cdot \bar{C} \cdot \bar{D} + A \cdot B \cdot C$. Pode-se notar que o termo $A \cdot B \cdot C$

¹ Alexandre César Rodrigues da Silva, DEE – FEIS - UNESP, Av. Brasil, 56, 15.385-000, Ilha Solteira, SP São Paulo, SP, Brasil, acrsilva@dee.feis.unesp.br

² Ivanil Sebastião Bonatti, FEEC - UNICAMP, Cidade Universitária Zeferino Vaz, 13081-970, Campinas, SP, Brazil, ivanil@dt.fee.unicamp.br

³ Cláudio Kitano, DEE – FEIS - UNESP, Av. Brasil, 56, 15.385-000, Ilha Solteira, SP São Paulo, SP, Brasil, kitano@dee.feis.unesp.br

cobre os termos produtos $A.B'.C'.D'$ e $A.B'.C'.D$, podendo ambos serem eliminados. A nova expressão torna-se $A'.B.C'.D + A'.B.C.D + A.B.C'.D + A.B'C'$. As operações de consenso e eliminação são empregadas tanto quanto possível. Com a aplicação sistemática tem-se $F_1(A.,B,C,D) = A.B'.C' + A.C'.D + A'.B.D + B.C'.D$, que representam os implicantes primos da função considerada. A Figura 1 apresenta no Mapa de Karnaugh os implicantes primos gerados.

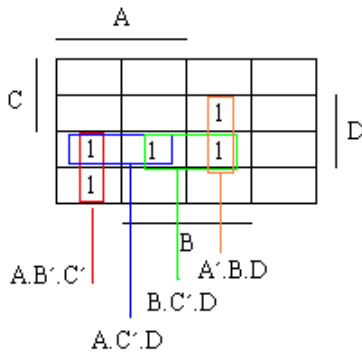


FIGURA. 1

MAPA DE KARNAUGH APRESENTANDO OS IMPLICANTES PRIMOS OBTIDOS PELA APLICAÇÃO DO CONSENSO INTERATIVO NA FUNÇÃO $F_1(A,B,C,D)$.

ALGORITMO PROPOSTO PARA A GERAÇÃO DOS IMPLICANTES PRIMOS

O algoritmo proposto denominado GeraPlex obtém todos os implicantes primos, de uma dada função booleana, utilizando-se da operação de consenso aplicada aos ramos de uma árvore de decisão binária que representa a função a ser minimizada.

Uma árvore de decisão binária é uma forma equivalente à Tabela Verdade utilizada para representar funções booleanas. Nesta estrutura os mintermos são representados pelos ramos da árvore. A Figura 2 apresenta a árvore de decisão para a função $F_2(A,B,C,D) = \sum m(0,1,3,5,9,13,15)$.

Note que na Figura 2 a raiz da árvore (variável A) representa o bit mais significativo e que as folhas (variável D) representa o bit menos significativo. Um mintermo (linha da Tabela Verdade cuja função assume valor lógico 1) é representado por um caminho da árvore, que vai da raiz à folha. Dessa forma, o caminho 1001 representa o mintermo 9.

Esta forma para representar funções booleanas é bastante conveniente para implementações computacionais, pois um mesmo ramo da árvore pode ser utilizado por mais de um mintermo, o que significa menos utilização de memória.

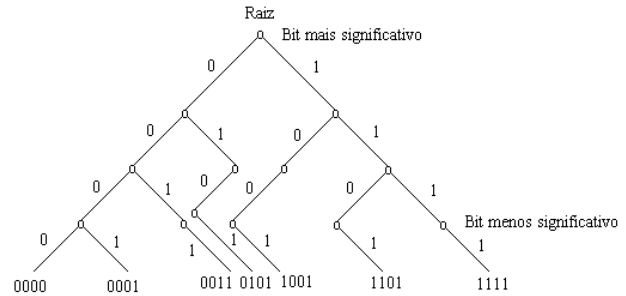


FIGURA. 2

ÁRVORE DE DECISÃO BINÁRIA PARA A FUNÇÃO $F_2(A,B,C,D)$.

No algoritmo GeraPlex, a geração dos implicantes primos tem início analisando-se a possibilidade de se aplicar o consenso aos ramos da árvore pertencentes aos nós do nível mais distante da raiz. Após aplicar o consenso aos ramos dos nós de um nível, avança-se para o próximo nível, no sentido da raiz. Desse modo, quando analisa-se um nó, todas as sub - árvores abaixo desse nó já devem ter sido analisadas. Da aplicação do consenso aos ramos da árvore são definidas três operações:

- **Fusão:** A operação de fusão ocorre quando dois ramos de um nó geram um terceiro ramo. O ramo gerado cobre os outros dois ramos do nó. Neste caso os ramos cobertos são eliminados. Esta operação diminui a quantidade de caminhos da árvore;
- **Deslocamento:** A operação de deslocamento ocorre quando dois ramos de um nó geram um terceiro ramo, porém o ramo gerado cobre apenas um dos dois outros ramos. Neste caso, o ramo coberto é eliminado. Esta operação mantém a quantidade de caminhos da árvore;
- **Expansão:** A operação de expansão ocorre quando dois ramos de um nó geram um terceiro ramo, porém o ramo gerado não cobre nenhum dos outros dois ramos.

Descrição Resumida do Algoritmo GeraPlex

O algoritmo GeraPlex obtém os implicantes primos de uma função booleana, e além disso, determina os implicantes primos essenciais, com vistas à formulação do problema de cobertura como um problema de programação matemática. Os seguintes passos compõe o algoritmo:

- **Montagem das árvores:** Monta-se duas árvores com a mesma estrutura; uma contendo somente os mintermos da função (árvore dos mintermos) e a outra contendo os mintermos e os irrelevantes (árvore dos implicantes). A árvore dos mintermos é utilizada no final do algoritmo para se determinar quais dos implicantes primos gerados são essenciais. Na árvore de implicantes são gerados os implicantes primos da função através das operações de fusão, deslocamento e expansão. Na lista ligada, em estrutura de árvore, cada registro tem três apontadores rotulados por “zero”, “um” e “dc”;

- **Geração dos implicantes primos:** Os implicantes primos da função são gerados na árvore de implicantes através de sucessivas aplicações do Método do Consenso aos ramos da árvore. Para evitar a geração de implicantes não primos deve-se inicialmente aplicar, para cada nó no mesmo nível da árvore, o consenso a ramos que produzem a fusão para depois aplicar o consenso a ramos que produzem o deslocamento ou a expansão. Para cada nível analisado deve-se eliminar os implicantes não primos. Após analisar o nó da raiz, a árvore de implicantes só contém caminhos que representam os implicantes primos da função;
- **Determinação dos implicantes primos essenciais:** Esse passo consiste em determinar os caminhos que representam os implicantes primos essenciais da função que devem ser eliminados da formulação do problema de cobertura objetivando a sua simplificação. No final desse procedimento a árvore de implicantes contém somente implicantes primos que não são essenciais. Isto corresponde, no Mapa de Karnaugh, a um caso cíclico, podendo ser formulado como um problema de programação linear inteira 0 e 1.
 - Seleciona-se um mintermo (caminho) da árvore de mintermos;
 - Verifica-se na árvore de implicantes quantos implicantes cobrem o mintermo selecionado;
 - Se existir apenas um implicante que cobre o mintermo selecionado, este é considerado essencial sendo removido da árvore de implicantes. Elimina-se também, da árvore de mintermos, todos os mintermos cobertos pelo implicante eliminado da árvore de implicantes;
 - Retorna-se ao início desse passo até que todos os mintermos tenham sido analisados.

Como exemplo de aplicação do algoritmo GeraPlex considere a função $F_3(A,B,C) = \sum m(0,1,5) + d(2,7)$, cujo Mapa de Karnaugh está apresentado na Figura 3.

		A	
		0	1
C	1	1	X
	0	X	1
		B	

FIGURA. 3
MAPA DE KARNAUGH DA FUNÇÃO $F_3(A,B,C)$.

A árvore inicial dos implicantes primos é mostrada na Figura 4. A geração dos implicantes primos inicia-se pela aplicação do consenso aos ramos dos nós do nível 0, ou seja, os mais distantes da raiz. A árvore apresentada na Figura 5 é o resultado da aplicação do consenso aos caminhos 000 e 001 que resulta no caminho 00X. Note que essa operação

corresponde à fusão, pois o caminho gerado cobre os outros dois.

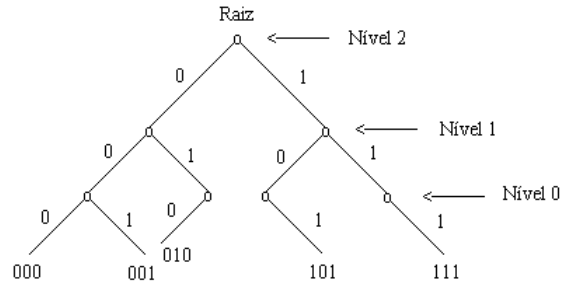


FIGURA. 4
ÁRVORE DE IMPLICANTES PRIMOS PARA A FUNÇÃO $F_3(A,B,C)$.

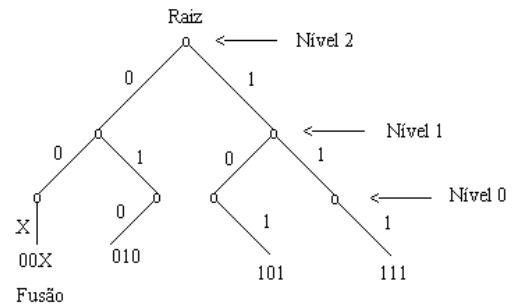


FIGURA. 5
ÁRVORE DE IMPLICANTES PRIMOS APÓS A APLICAÇÃO DO CONSENSO AOS NÓS DO NÍVEL 0.

Após a aplicação do consenso aos ramos de um nível da árvore, avança-se para os nós do próximo nível no sentido da raiz. A árvore apresentada na Figura 6 é o resultado da aplicação do consenso aos ramos dos nós de nível 1. O consenso entre os caminhos 101 e 111 gera o caminho 1X1, que cobre os outros dois caminhos. Observe que o consenso entre os caminhos 00X e 010 gera o caminho 0X0, que cobre somente o caminho 010. Esta operação é definida como deslocamento.

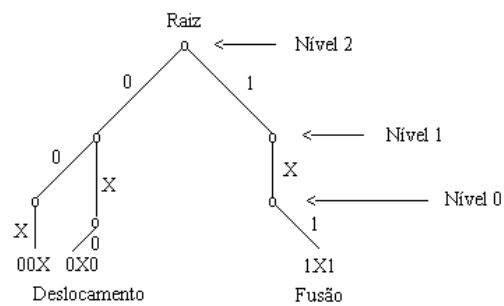


FIGURA. 6
ÁRVORE DE IMPLICANTES PRIMOS APÓS A APLICAÇÃO DO CONSENSO AOS NÓS DO NÍVEL 1.

Para o nó de nível 2 (raiz), da árvore apresentada na Figura 6, o consenso aplicado aos caminhos 00X e 1X1 gera o caminho X01. O caminho gerado não cobre nenhum dos outros dois caminhos. Essa operação é chamada de expansão, pois aumenta o quantidade de caminhos na árvore, como pode ser observado na Figura 7.

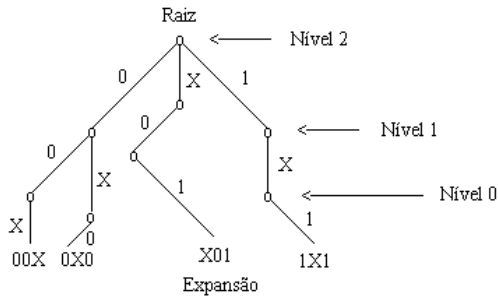


FIGURA. 7

ÁRVORE DE IMPLICANTES PRIMOS APÓS A APLICAÇÃO DO CONSENSO AO NÓ DO NÍVEL 2.

Observe que na árvore apresentada na Figura 7 não se pode mais aplicar o consenso. Todos os caminhos da árvore representam implicantes primos da função como mostra o Mapa de Karnaugh apresentado na Figura 8.

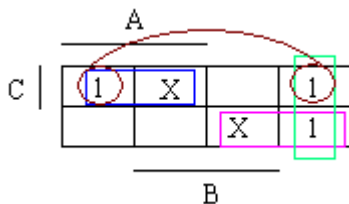


FIGURA. 8

MAPA DE KARNAUGH COM OS IMPLICANTES PRIMOS GERADOS PELO PROGRAMA GERA PLEX PARA A FUNÇÃO F₃.

Com os implicantes primos obtidos, o problema de cobertura de funções booleanas pode ser formulado como um problema de programação linear inteira 0 e 1. Para a função F₃ tem-se a seguinte formulação do problema de cobertura:

$$\begin{aligned} \text{Min } F_3 &= 3.1X1 + 3.0X0 + 3.00X + 3.X01 \\ \text{s.a.} \\ 0X0 + 00X &\geq 1 \\ 00X + X01 &\geq 1 \\ 1X1 + X01 &\geq 1 \\ 1X1, 0X0, 00X \text{ e } X01 &\in \{0, 1\} \end{aligned}$$

Pode-se notar que o problema de cobertura passou a ser tratado como um problema de programação matemática, podendo-se, dessa forma, utilizar-se de todos os avanços dessa área de pesquisa.

Para as dezenas de casos estudados, o programa GeraPlex obteve os mesmos implicantes primos obtidos pelo método tabular de Quine-McCluskey. Foram também estudadas funções denominadas Tudo Um, isto é, funções contendo todos os mintermos possíveis. Trata-se de uma função sem uso prático, entretanto, consiste em um bom teste para avaliar o desempenho de algoritmos, pois exige grande quantidade de memória e consome muito tempo de computação.

No início do desenvolvimento desta pesquisa, pretendia-se comparar a eficiência do algoritmo desenvolvido com outros algoritmos implementados em pesquisas anteriores. Estas comparações foram realizadas, entretanto, foram logo desconsideradas, pois entendeu-se serem comparações desleais. Dois foram os motivos. Primeiro, as estruturas de dados utilizadas no programa GeraPlex são muito diferentes das estruturas empregadas no Quine-McCluskey e nos outros algoritmos, e com toda certeza essa foi a principal vantagem do GeraPlex sobre os outros algoritmos avaliados. Segundo, as linguagens de programação utilizadas foram diferentes, o que torna sem sentido qualquer tipo de comparação em termos de tempo de execução e uso de memória dos algoritmos.

O programa GeraPlex estará sendo comparado com um outro programa para a geração de implicantes primos que implementa o Método de Expansão de Shannon. Este outro programa, denominado Expander também utiliza-se de estrutura em diagrama de decisão binária e está programado na linguagem de programação C, da mesma forma que o GeraPlex.

Este é o motivo pelo qual não apresentou-se os dados comparativos entre o algoritmo implementado e alguns outros método clássicos. Entretanto, pode-se afirmar com muita segurança que a abordagem dada neste trabalho consiste em uma forma bastante eficiente para a geração de implicantes primos e formulação do problema de cobertura de funções booleanas como um problema de programação matemática.

CONCLUSÃO

Este trabalho apresentou uma alternativa eficiente para a geração de todos os implicantes primos de uma função booleana, a partir dos quais o problema de cobertura pode ser formulado como um problema de programação linear inteira 0 e 1.

Os implicantes primos da função foram gerados aplicando-se a operação de consenso aos ramos de uma árvore de decisão binária cujos caminhos (raiz até folha) representam os mintermos e irrelevantes da função. Da aplicação do consenso aos ramos da árvore são definidas as operações de fusão, deslocamento e expansão, que diminui, mantém e aumenta, respectivamente, os caminhos da árvore. No final do processo os caminhos que permanecem na árvore representam os implicantes primos da função, que no

Método de Quine-McCluskey (tradicional método tabular) corresponde a um Mapa Cíclico.

Dezenas de funções booleanas foram minimizadas e os resultados comparados com tradicionais métodos de geração de implicantes primos. Em todos os casos o GeraPlex obteve os mesmos implicantes primos e apresentou desempenho superior em tempo e memória.

AGRADECIMENTOS

Os autores agradecem o apoio recebido do PPGEE da Faculdade de Engenharia Elétrica de Ilha Solteira – UNESP e da CAPES.

REFERÊNCIA

- [1] R. J. Nelson, “Simplest Normal Truth Function”, *Journal of Symbolic Logic*, Vol. 20, June, 1954, pp.105.
- [2] B. L. Hulme, R., B., Worrel, “A Prime Implicant Algorithm with Factoring”, *IEEE Trans. on Computers*, November, 1975, pp. 1129.
- [3] A. H. Scheinman, “A Method for Simplifying Boolean Functions”, *The Bell System Technical Journal*, July, 1962, pp. 1336.
- [4] R. S. Das, A., K., Choudhury, “Maxterm Type Expressions of Switching Functions and their Prime Implicants”, *IEEE Trans. Electronic Computers*, Vol. EC-14, December, 1965, pp. 920.
- [5] J. R. Slagle, C., L., Chang, R., C., T., Lee, “A New Algorithm for Generating Prime Implicants”, *IEEE Trans. Computers*, Vol. C-19, April, 1970, pp. 304.
- [6] W. V. Quine, “The Problem of Simplifying Truth Function”, *American Mathematical Monthly*, Vol. 59, October, 1952, p. 521.
- [7] W. V. Quine, “The Way to Simplify Truth Function”, *American Mathematical Monthly*, Vol. 62, November, 1955, p. 627.
- [8] P. Tison, “Generalization of Consensus Theory and Application to Mimization of Boolean Function”, *IEEE Trans. Electronic Computer*, Vol. EC-16, August, 1967, p. 446.
- [9] G. De Micheli, “Synthesis and Optimization of Digital Circuits”, *McGraw Hill Inc*, 1994.
- [10] E. Mendelson, “Boolean Algebra and Switching Circuits: Theory and Problems”, *McGraw Hill Inc*, 1970.