PROGRAMAÇÃO MATEMÁTICA APLICADA NO PROBLEMA DE COBERTURA DE FUNÇÕES BOOLEANAS COM MÚLTIPLAS SAÍDAS

Alexandre César Rodrigues da Silva[1], *Ivanil Sebastião Bonatti*[2] *e Cláudio Kitano*[3]

Resumo — *A complexidade dos sistemas digitais exige circuitos contendo várias saídas, ao invés de apenas uma. Essas funções de saídas podem ser implementadas independentemente, porém uma considerável economia ocorrerá se forem implementadas como um todo. Essa economia decorre do uso de um mesmo termo produto por duas ou mais funções, que é implementado uma única vez e compartilhado pelas funções. Apresenta-se neste trabalho o algoritmo MultPlex, que minimiza funções booleanas com múltiplas saída. Para a fase de geração de implicantes primos, utilizou-se um tradicional método tabular, que foi estendido para considerar as múltiplas funções. As Coberturas das funções são resultado de dois problemas de programação matemática, aqui denominados de problema de cobertura múltipla e problema de cobertura singular. Os testes realizados mostram que este tipo de enfoque resolve o problema de minimização de funções com múltiplas saídas de maneira eficiente e representa um avanço considerável para esta classe de problema.*

Palavras chave — Minimização de funções Booleanas, Otimização, Programação matemática, Síntese lógica .

INTRODUÇÃO

O problema de simplificar funções booleanas que representam circuitos digitais com um grande número de variáveis e com múltiplas saídas, onde as saídas dependem das mesmas variáveis de entrada, ainda continua sem uma solução satisfatória. Seguramente as funções para as múltiplas saídas podem ser minimizadas separadamente, contudo esse procedimento não leva a uma solução de menor custo.

Tanto o Mapa de Karnaugh quanto o Método de Quine-McCluskey [1] podem ser estendidos para simplificar várias funções simultaneamente, através do compartilhamento dos termos comuns. Tal procedimento torna-se impraticável se o número de variáveis ou o número de funções forem grandes.

Uma grande quantidade de algoritmos como, por exemplo, os descritos em [2] - [3] - [4] - [5] - [6] e [7] foram desenvolvidos na tentativa de solucionar esse problema. Um algoritmo denominado McBoole [8] obtém uma cobertura mínima para funções booleanas com múltiplas saídas expressa como uma lista de cubos. Baseando-se na manipulação eficiente de grafos e em técnicas de partição, o McBoole encontra, quase sempre, uma cobertura mínima global. Trata-se de um algoritmo atrativo para funções com até 20 variáveis de entrada e até 20 funções de saída.

Neste trabalho formulou-se o problema de minimização de funções booleanas com múltiplas saídas como um problema de programação linear inteira 0 e 1 e apresenta-se um algoritmo denominado MultPlex, que obtém a cobertura mínima, resolvendo os problemas matemáticos formulados. O critério de custo adotado é o número de portas ANDs na realização das funções.

Para a geração dos implicantes primos, utilizou-se um método tabular, adaptado para considerar funções com múltiplas saídas.

As coberturas das funções de saídas são soluções de dois problemas de programação matemática. No primeiro problema, denominado de cobertura múltipla, formulou-se o problema de cobertura como um problema de programação matemática cujas variáveis são os implicantes primos singulares (pertencentes a uma única função) e os implicantes primos múltiplos (pertencentes a uma ou mais funções) das funções de saída. No problema matemático gera-se uma restrição de cobertura para cada mintermo de todas as funções de saídas.

No segundo problema, denominado de cobertura singular, tem-se como variáveis os termos solução do problema de cobertura múltipla. Desse modo, tem-se tantos problemas singulares quanto forem o número de saídas da função sob estudo.

Na função objetivo de cada problema considera-se somente os implicantes primos solução da cobertura múltipla que cobre a função em questão. Nas restrições de cobertura, considera-se somente os mintermos da função.

GERAÇÃO DE IMPLICANTES PRIMOS PARA FUNÇÕES COM MÚLTIPLAS SAÍDAS

Para a cobertura de funções booleanas com múltiplas saídas considera-se, além dos implicantes primos singulares (que cobrem mintermos pertencentes a apenas uma das funções), os implicantes primos múltiplos (que cobrem mintermos originados do produto lógico de duas ou mais funções de saída).

Na geração dos implicantes primos utilizou-se o tradicional Método Tabular de Quine-McCluskey [1] que foi

[1] Alexandre César Rodrigues da Silva, DEE – FEIS - UNESP, Av. Brasil, 56, 15.385-000, Ilha Solteira, SP, Brasil, acrsilva@dee.feis.unesp.br
[2] Ivanil Sebastião Bonatti, FEEC - UNICAMP, Cidade Universitária Zeferino Vaz, 13081-970, Campinas, SP, Brasil, ivanil@dt.fee.unicamp.br
[3] Cláudio Kitano, DEE – FEIS - UNESP, Av. Brasil, 56, 15.385-000, Ilha Solteira, SP, Brasil, kitano@dee.feis.unesp.br

estendido para considerar funções com múltiplas saídas. Para esse tipo de funções os mintermos e os irrelevantes devem conter, além do identificador (representação binária de um mintermo ou irrelevante), uma outra seqüência de bits, denominada de rótulo, que assinala quais funções de saída incluem o termo produto (mintermo ou irrelevante). Desse modo, a notação 0100 11 significa que o mintermo 4 que é representado por $(0100)_2$ pertence as funções F_a e F_b representadas pela notação 11. Da mesma forma, a notação 11X0 01 significa que o implicante 11X0 cobre somente o mintermo pertencente à função F_b, pois tem-se 0 no identificador da função F_a.

Para a geração dos implicantes primos singulares e implicantes primos múltiplos, os mintermos e os irrelevantes são combinados da mesma forma que no tradicional Método de Quine-McCluskey, contudo algumas observações tornam-se necessárias:

- Só podem ser combinados mintermos ou irrelevantes comuns (assinalados para as mesmas funções de saída);
- Quando dois mintermos ou irrelevantes combinarem-se, os bits do rótulo do implicante resultante terá o dígito 1, somente nas posições onde ambos os rótulos dos mintermos ou irrelevantes combinados têm 1, ou seja, aplica-se a operação AND entre os bits dos rótulos. Por exemplo, a combinação do mintermo 0110 10 com o mintermo 1110 11 gera o implicante X110 10, que pertence somente à função f_a.
- Marcar um implicante como tendo sido combinado, somente se seu rótulo está contido no rótulo do implicante resultante da combinação.

Para exemplificar o emprego do método tabular para a geração de implicantes primos para múltiplas funções considere as funções $F_1(A,B,C) = \Sigma m(0,1,3,5)$, $F_2(A,B,C) = \Sigma m(2,3,5,6)$ e $F_3(A,B,C) = \Sigma m(0,1,6)$. A Figura 1 apresenta a tabela para a geração dos implicantes primos.

	A B C	f1 f2 f3	
0	0 0 0	1 0 1	*
1	0 0 1	1 0 1	*
2	0 1 0	0 1 0	*
3	0 1 1	1 1 0	
5	1 0 1	1 1 0	
6	1 1 0	0 1 1	
0-1	0 0 X	1 0 1	
1-3	0 X 1	1 0 0	
1-5	X 0 1	1 0 0	
2-3	0 1 X	0 1 0	
2-6	X 1 0	0 1 0	

Identificador Rótulo

FIGURA. 1

TABELA DE GERAÇÃO DE IMPLICANTES PRIMOS PARA AS FUNÇÕES F_1, F_2 , F_3.

O procedimento para a seleção dos implicantes primos, que corresponde à uma cobertura mínima para as funções em análise, pode ser o mesmo utilizado para a cobertura de uma única função. Entretanto, este método, que já é bastante complexo para tratar de uma única função, torna-se extremamente penoso e ineficiente para trabalhar com múltiplas funções.

Abordagem dada neste trabalho foi a formulação do problema de cobertura múltiplas funções de saída como um problema de programação linear inteira 0 e 1.

COBERTURA DE FUNÇÕES BOOLEANAS COM MÚLTIPLAS SAÍDAS SOB A ÓPTICA DA PROGRAMAÇÃO MATEMÁTICA

O problema de cobertura de funções booleanas com múltiplas saídas pode ser formulado como um problema de programação matemática.

Adotou-se como critério de custo a quantidade de portas ANDs necessárias para a implementação das múltiplas funções. Dessa forma, o problema de cobertura das múltiplas funções de saída torna-se bastante similar ao problema de cobertura para uma única função de saída. A diferença é que, no caso de funções com múltiplas saídas, a cobertura é efetuada em duas etapas.

Na primeira etapa, denominada de cobertura múltipla, as variáveis do problema matemático, são os implicantes primos singulares e os implicantes primos múltiplos, obtidos pelo método de geração de implicantes primos. As restrições de cobertura são tantas quanto forem os mintermos de todas as funções de saída. Se um mintermo está incluído em duas funções, tem-se duas restrições para esse mintermo. Apresenta-se a seguir a formulação do problema de cobertura múltipla para as funções F_1, F_2 e F_3.

$$
\begin{aligned}
&\text{Min } F = 3.(00X+0X1+X01+01X+X10) + 4.(011+101+110)\\
&\quad \text{s.a}\\
&\qquad 00X \geq 1\\
&\qquad 00X + 0X1 + X01 \geq 1\\
&\qquad 011 + 0X1 \geq 1\\
&\qquad 101 + X01 \geq 1\\
&\qquad 01X + X10 \geq 1\\
&\qquad 011 + 01X \geq 1\\
&\qquad 101 \geq 1\\
&\qquad 110 + X10 \geq 1\\
&\qquad 00X \geq 1\\
&\qquad 00X \geq 1\\
&\qquad 110 \geq 1\\
&\qquad 00X, 0X1, X01, 01X, X10, 011, 101, 110 \in \{0, 1\}
\end{aligned}
$$

Pode-se notar que o critério de custo é dado pela soma ponderada dos implicantes primos múltiplos e dos implicantes primos singulares e que há uma restrição para cada mintermo de todas as funções de saída. Desse modo, as quatro primeiras restrições referem-se, respectivamente, à cobertura dos mintermos 0, 1, 3 e 5 de F_1. As próximas quatro restrições se referem aos mintermos 2, 3, 5 e 6 de F_2 e as três últimas restrições referem-se aos mintermos 0, 1 e 6 de F_3. As restrições redundantes podem ser eliminadas.

A solução da cobertura múltipla é obtida utilizando-se algum método matemático de resolução de programação matemática como, por exemplo, o Método Simplex e o Plano de Corte de Gomory [9]. A solução para o problema em questão, apresentado como exemplo, é dada pelos implicantes 101 110, 110 011, 00X 101, 0X1 100 e 01X 010.

Obtida a solução da cobertura múltipla, inicia-se a segunda etapa da cobertura. Na segunda etapa, denominada de cobertura singular, tem-se um problema de programação matemática para cada uma das funções de saída, sendo que as variáveis do problema são os implicantes primos, solução da primeira etapa, que cobrem as funções em questão. As coberturas singulares para as funções F_1, F_2 e F_3 são de solução trivial. A função F_1 é coberta pelos implicantes 101, 00X e 01X, cujo custo dos ANDs é igual a 7. A função F_2 é coberta pelos implicantes primos 101, 110 e 01X, cujo custo é igual a 8, e a função F_3 é coberta pelos implicantes primos 110 e 00X, cujo custo é igual a 5.

O custo total dos ANDs, para realizar as funções F_1, F_2 e F_3, como uma função de múltiplas saídas é igual a 12, pois os mintermos 101 e 110 e o implicante primo 00X aparecem duas vezes na função. O custo do circuito, ANDs e OR, é igual a 12 + 3 + 3 + 2 = 20. As três funções quando implementadas independentemente têm um custo total igual a 26. Nota-se, portanto, uma economia de 6 unidades de custo, quando funções booleanas são tratadas como funções de múltiplas saídas.

O Algoritmo MultPlex

O algoritmo MultPlex, que obtém uma cobertura de custo mínimo para uma função booleana de múltiplas saídas, consiste nos seguintes passos:

1. Geração dos implicantes primos pelo método tabular;
2. Montagem do Tableau;
3. Obtenção de uma base inicial, relaxando-se as restrições de integralidade;
4. Verificação da factibilidade da solução básica inicial obtida;
5. Obtenção de uma base inicial factível , se a base obtida não o for;
6. Minimização da função objetivo;
7. Reconsiderando-se a integralidade das variáveis, faz-se a verificação da factibilidade da solução ótima obtida;
8. Obtenção de uma solução inteira, através do Plano de Corte de Gomory, se a solução obtida não o for;
9. Geração da cobertura singular, para cada uma das funções de saída;
10. Executar os passos de 2 até 8 para cada uma das formulações de cobertura singular geradas no passo anterior.

Como exemplo de aplicação do algoritmo descrito, considere as funções booleanas $F_4(A,B,C,D) = \Sigma m(2,3,6,10) + d(8)$, $F_5(A,B,C,D) = \Sigma m(2,10,12,14) + d(6,8)$ e $F_6(A,B,C,D) = \Sigma m(2,8,10,12) + d(0,14)$.

A primeira fase do método gera os seguintes implicantes primos: 0X10 110, 001X 100, X010 111, 10X0 111, X0X0 001, XX10 010 E 1XX0 011.

A formulação para a cobertura múltipla é dada por:

$$\text{Min } F = 3.(X0X0+XX10+1XX0) + 4.(0X10+001X+X010+10X0)$$

s.a

$$0X10 + 001X + X010 \geq 1$$
$$001X \geq 1$$
$$0X10 \geq 1$$
$$X010 + 10X0 \geq 1$$
$$0X10 + X010 + XX10 \geq 1$$
$$X010 + 10X0 + XX10 + 1XX0 \geq 1$$
$$1XX0 \geq 1$$
$$XX10 + 1XX0 \geq 1$$
$$X010 + X0X0 \geq 1$$
$$10X0 + X0X0 + 1XX0 \geq 1$$
$$X010 + 10X0 + X0X0 \geq 1$$
$$1XX0 \geq 1$$
$$X0X0, XX10, 1XX0, 0X10, 001X, X010, 10X0 \in \{0, 1\}$$

A solução ótima, para esse problema de programação linear, é dada por: 0X10 110, 001X 100, 1XX0 011 e X010 111. As coberturas das funções para a solução obtida são apresentadas no Mapa de Karnaugh da Figura 2.

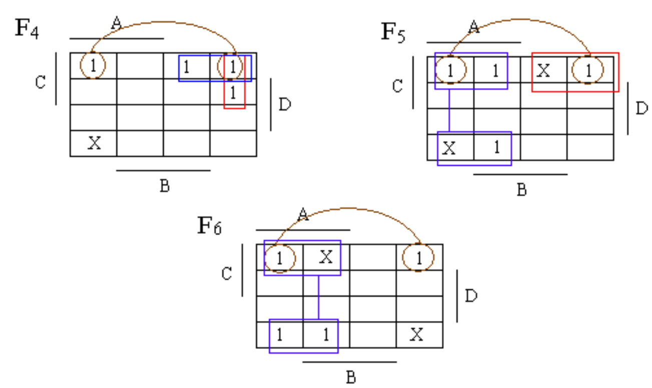

FIGURA. 2
COBERTURAS PARA AS FUNÇÕES BOOLEANAS F_4, F_5 E F_6 APRESENTADAS NO MAPA DE KARNAUGH.

Pode-se notar, pelo Mapa de Karnaugh da função F_5 que o implicante primo 0X10 é redundante, pois cobre mintermos já coberto por outro implicante primo. Portanto, após obter a solução para a cobertura múltipla, deve-se fazer uma cobertura singular, uma para cada função de saída, cujas variáveis são os implicantes primos obtidos como solução da cobertura múltipla.

A cobertura singular para a função F_4 é dada por:

$$\text{Min } F_4 = 4.(0X10 + 001X + X010)$$

s.a

$$0X10 + 001X + X010 \geq 1$$
$$001X \geq 1$$
$$0X10 \geq 1$$
$$X010 + 10X0 \geq 1$$
$$0X10,\ 001X \text{ e } X010 \in \{0, 1\}$$

A solução para este problema é dada por: 001X, 0X10 e X010.

A cobertura singular para a função F_5 é dada por:

$$\text{Min } F_5 = 3.\ 1XX0 + 4.(0X10 + X010)$$

s.a

$$0X10 + X010 \geq 1$$
$$1XX0 + X010 \geq 1$$
$$1XX0 \geq 1$$
$$1XX0,\ 0X10,\ X010 \in \{0, 1\}$$

A solução para este problema é dada por: 1XX0 e X010.

Note que o implicante primo 0X10, pertencente a solução da cobertura múltipla, foi eliminado da cobertura da função F_5, pois esse implicante é essencial somente para a cobertura da função F_4.

A cobertura singular para a função F_6 é dada por:

$$\text{Min } F_6 = 3.\ 1XX0 + 4.\ X010$$

s.a

$$X010 \geq 1$$
$$1XX0 \geq 1$$
$$1XX0 + X010 \geq 1$$
$$1XX0 \text{ e } X010 \in \{0, 1\}$$

A solução para este problema é dada por: 1XX0 e X010.

O custo para a realização das ANDs das três funções, como uma função de múltiplas saídas, corresponde a 11, enquanto o custo das ANDs para a realização das três funções, como funções independentes, corresponde a 17.

Muitas outras funções de uso prático como, por exemplo, o código de linha HDB-3 [10], foram implementadas utilizando-se o algoritmo desenvolvido neste trabalho. O problema de cobertura para o HDB-3, considerado de grande porte, gerou um problema matemático contendo 87 variáveis e 215 restrições. Vale mencionar que o MultPlex obteve os mesmos resultados gerados pelo programa Espresso [8].

Conclusão

Esta pesquisa tratou da minimização de funções booleanas que implementam circuitos digitais com múltiplas saídas, evidenciando a necessidade de se considerar esse tipo de implementação em projetos de sistemas digitais.

O algoritmo MultPlex foi apresentado através de alguns exemplos. Para a fase de geração dos implicantes primos utilizou-se um método tabular, modificado para considerar funções booleanas com múltiplas saídas. Obtido os implicantes primos das funções o problema de cobertura é formulado como um problema de programação matemática.

A fase de cobertura é efetuada em duas etapas, denominadas de cobertura múltipla e cobertura singular.

Na cobertura múltipla considera-se as funções de saída como um todo e formula-se um problema de programação linear inteira com os implicantes primos múltiplos (implicantes pertencentes a uma ou mais funções) e os implicantes primos singulares (pertencente a apenas uma das funções). Com o resultado da cobertura múltipla formulam-se as coberturas singulares, onde as funções de saída são consideradas isoladamente.

O algoritmo MultPlex representa uma contribuição definitiva na simplificação de funções booleanas de múltiplas saídas, pois define o problema de simplificação como um problema de programação matemática. Dessa forma, métodos computacionais eficazes podem ser empregados com o objetivo de solucioná-lo.

Agradecimentos

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

Referências

[1] E. J. McCluskey, "Logic Design Principles with Emphasis on Testable Semicustom Circuits", *Prentice Hall,* N.J., 1986.

[2] S. R. Perkins, T. Rhyne, "An Algorithm for Identifying and Selecting the Prime Implicants of a Multiple-Output Boolean Function", *IEEE Trans. on Computers-Aided Design*, Vol. 7, No 11, November, 1988, pp. 1215.

[3] S. J. Hong, R. G. Cain, D. L. Ostapko, "MINI: A Heuristic Approach for Logic Minimization", *IBM Journal Res. Development*, September, 1974, pp.443.

[4] Z. Arevalo, J. G. Bredeson, "A Method to Simplify a Boolean Function in a Near Minimal Sum-of-Product for Programmable Logic Array", *IEEE Trans. on Computers*, Vol. C-27, November, 1978, pp. 1028.

[5] K. B. Brayton, G. D. Hachtel, C. T. McMullen, ª L. Sangiovanni-Vicentelli, "Logic Minimization Algoritms for VLSI Synthesis", *Kluwer Academic Publisher*, 1984.

[6] K. Karplus, "Using if-then-else DAGs for Multi-Level Logic Minimization", *UCSC-CRL-88-29, University of California*, November, 1988.

[7] B. Gurunath, N. N. Biswas, “An Algorithm for Multiple Output Minimization”, *IEEE Trans. on Computer-Aided-Design*, Vol. 8, No. 9, September, 1989, pp.1007.

[8] M. R. Dagenais, V. K. Agarwal, N. C. Rumin, “McBoole: A New Procedure for Exact Logic Minimization”, *IEEE Trans. on Computer-AidedDesign*, Vol. CAD-5, No. 1, January, 1986, pp.229

[9] A. C. R. da Silva, ”Contribuição à Síntese de Circuitos Digitais Utilizando Programação Linear Inteira 0 e 1”, *Tese de Doutorado*, UNICAMP, Setembro, 1993, pp. 127.

[10] D. B. Keogh, “The State Diagram of HDB-3”, *IEEE Trans. on Communications*, Vol. Com-32, No. 11, November, 1984, pp.1222.