

COMPARAÇÃO DE MÉTODOS DE SÍNTESE DE FUNÇÕES BOOLEANAS

Gabriel Ammes e Renato P. Ribas

Instituto de Informática

{gabriel.ammes, renato.ribas}@inf.ufrgs.br

RESUMO

Este trabalho apresenta uma avaliação de métodos de síntese de funções Booleanas considerando métricas como número de literais, número de operações lógicas, profundidade lógica do circuito e tempo de execução. Existem diversas abordagens para síntese de funções lógicas, mas a relação entre elas não é bem definida. Esta análise se torna útil na decisão de qual método é o mais indicado para uma dada aplicação.

MÉTODOS DE SÍNTESE

Tabela Verdade de uma dada função Booleana, como um estudo de caso:

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Expressão de dois níveis POS/SOP: Esta é a representação mais direta de uma função Booleana em formato de expressão, em relação a uma dada tabela verdade.

Uma Soma-de-Produtos de literais (SOP) ou cubos:

$$F = A!B!C + A!B!C + A!B!C + !A!B!C$$

Um Produto-de-Somas de literais (POS) ou cláusulas:

$$F = A+B+C * A+B+!C * A+!B+C * !A+!B+!C$$

Expressão baseada em MUX: O comportamento de um bloco multiplexador (MUX) pode ser logicamente representado por uma SOP. Usando uma variável como entrada, é possível minimizar a SOP original.

$$F = !A!B!C + A!B + A!B!C$$

Método de Quine-McCluskey: Este método é utilizado para encontrar a SOP com o menor número de cubos.

$$F = A!B + !A!B!C + A!C$$

Método de Quine-McCluskey utilizando XOR: Esta é uma modificação ao método de Quine-McCluskey com objetivo de considerar operação Booleana OU-exclusivo (XOR) nos cubos

$$F = C!(B^A) + (A!C)$$

Expressão XOR: Para minimizar uma expressão utilizando portas XOR, as seguintes expansões podem ser aplicadas:

$$\begin{matrix} \text{Davio positivo} & \text{Davio negativo} & \text{Shannon} \\ F = 1*f_0 \wedge x_1*f_2 & F = !x_1*f_2 \wedge 1*f_1 & F = !x_1*f_0 \wedge x_1*f_1 \end{matrix}$$

Expressão Reed-Muller de Polaridade Positiva: A expansão de Davio positiva é aplicada para cada variável.

$$F = A^B(A^A)^{C^B(A^B(A^A))^A^B}$$

Expressão Reed-Muller de Polaridade Fixa: É aplicada a expansão de Davio Positivo ou Negativo para cada variável.

$$F = A \wedge (C^B(A^B))$$

Expressão Kronecker: Além da aplicação da expansão de Davio positiva ou negativa, também considera a expansão de Shannon para cada variável.

$$F = (!C^A) \wedge (C^B(A^B))$$

Expressão Pseudo Reed-Muller: Aplica a expansão de Davio positiva ou negativa para cada variável e para cada sub-expressão.

$$F = A \wedge (B^C)$$

Expressão Pseudo Kronecker: Aplica a expansão de Davio positiva ou negativa ou a expansão de Shannon para cada variável e para cada sub-expressão.

$$F = A \wedge (B^C)$$

RESULTADOS EXPERIMENTAIS

Estes métodos foram implementados utilizando a linguagem de programação Java e os resultados foram validados utilizando o framework de síntese lógica Karma. Os resultados experimentais para todas as funções de 3 variáveis são mostrados na Tabela I, enquanto que os resultados para mil funções de 5 variáveis geradas aleatoriamente são mostrados na Tabela II. O método para expressões Pseudo Kronecker pode ser aplicado em funções de até 3 variáveis e o de expressões Pseudo Reed-Muller em funções de até 4 variáveis.

TABELA I: Todas funções Booleanas de 3 variáveis.

Método	Literais	AND/OR	XOR	Profundidade	Tempo (s)
SOP/POS	12.0	11.0	0.0	2.0	0.01
MUX	6.7	5.7	0.0	1.9	0.02
QM	5.0	4.0	0.0	1.8	0.15
QMXOR	3.8	2.2	0.6	2.0	0.19
ERMPP	6.0	2.4	2.6	3.3	0.09
ERMPF	4.9	2.5	1.8	3.0	0.23
KRO	4.5	2.5	1.0	2.0	0.32
EPSDRM	4.0	1.8	1.2	1.7	0.32
PSDKRO	4.0	1.8	1.1	1.7	2.80

TABELA II: Funções Booleanas aleatórias de 5 variáveis.

Método	Literais	AND/OR	XOR	Profundidade	Tempo (s)
SOP	59.6	58.6	0.0	2.0	0.03
POS	101.3	100.3	0.0	2.0	0.04
MUX	38.3	37.3	0.0	2.0	0.15
QM	26.5	25.5	0.0	2.0	0.45
QMXOR	15.5	11.4	3.1	2.8	1.77
ERMPP	50.3	23.0	26.2	7.7	1.20
ERMPF	38.4	18.9	18.5	7.5	5.50
KRO	22.1	14.1	7.0	6.8	29.00