Upper Bound on the Number of Products in the AND-OR-XOR Expansion of Logic Functions.

E.V. Dubrova D.M. Miller J.C. Muzio

VLSI Design and Test Group
Department of Computer Science
University of Victoria, P.O.Box 3055
Victoria, B.C., Canada, V8W 3P6

Abstract

The representation of logic functions consisting of an XOR sum of two sum-of-products expressions is considered. The upper bound on the number of products in such the representation is shown to be $5 \cdot 2^{n-4}$, which is 37.50% smaller than the upper bound for sum-of-product expression and 16.67% smaller than the bound for AND-XOR expression.

1 Introduction

In 1990 Sasao and Besslich [4] showed that the upper bound on the number of products in an AND-XOR expansion in which both complemented and uncomplemented forms of a variable are used is $3 \cdot 2^{n-3}$. This bound is 25% less than the upper bound on the number of products in the conventional sum-of-product expression of logic functions, which is clearly 2^{n-1}.

However, in some technologies, the implementation of an XOR operation is more complex than the implementation of an OR operation. For example, in fully complemented CMOS an XOR gate takes more chip area than an OR gate (12 versus 6 transistors) and has the larger delay. So, even if a logic function has fewer products in its AND-XOR expression than in its sum-of-product expression, it does not guarantee that its CMOS AND-XOR implementation will occupy less area than its AND-OR implementation.

We show that the representation of logic functions as an XOR sum of two sum-of-products expressions has a lower upper bound on the number of products than an AND-XOR expansion. Since only one XOR operation per function is used, such an expansion could have an advantage over an AND-XOR expansion for technologies like CMOS, in which an XOR operation has a more complex implementation than an OR operation. Of course, there is a speed tradeoff as the proposed expansion has three levels.

2 AND-OR-XOR expansion

In this section we present a theorem which gives the upper bound on the number of products in the representation of logic functions consisting of an XOR sum of two sum-of-products expressions.

Let f_1, f_2, h_1, h_2 denote Boolean functions and x denote a Boolean variable. We use "$\cdot$" for AND, "$+$" for OR and "$\oplus$" for XOR operations. "$\cdot$" is omitted when obvious.

Property 1 *The following equations hold:*

$$\overline{x}(f_1 \oplus f_2) + x(h_1 \oplus h_2) = (\overline{x}f_1 + xh_1) \oplus (\overline{x}f_2 + xh_2)$$

$$\overline{x}(f_1 \oplus f_2) + x(h_1 \oplus h_2) = (\overline{x}f_1 + xh_2) \oplus (\overline{x}f_2 + xh_1)$$

Proof: The proof is straightforward.

□

Let P_i and Q_i, $i > 0$, denote arbitrary products involving some of the variables $x_1, \ldots, x_n$ or their complements. For example, the function $f(x_1, x_2, x_3) = x_1x_2 + \overline{x}_1x_2\overline{x}_3$ can be written as $f(x_1, x_2, x_3) = P_1 + P_2$, with $P_1 = x_1x_2$ and $P_2 = \overline{x}_1x_2\overline{x}_3$.

Theorem 1 *Every Boolean function $f(x_1, \ldots, x_n)$ of n variables ($n \geq 4$) can be expanded using at most $5 \cdot 2^{n-4}$ products as:*

$$f(x_1, \ldots, x_n) = (P_1 + P_2 + \ldots + P_p) \oplus (P_{p+1} + P_{p+2} + \ldots + P_m),$$

for some p in $1 \leq p \leq m$.

Proof: By induction on n.

1) Let $n = 4$. By exhaustive search through 402 PN-equivalence classes [1] of Boolean functions of 4-variables we can establish that each function can be expanded using at most 5 product terms. Hence, for $n = 4$ the expansion exists and $m \leq 5$.

2) Hypothesis: Assume the result holds for functions of n or less variables. Using a Shannon decomposition, any Boolean function of $n+1$ variables can be expanded as:

$$f(x_1, \ldots, x_{n+1}) = \overline{x}_{n+1}f_0(x_1, \ldots, x_n) + x_{n+1}f_1(x_1, \ldots, x_n),$$

[1] In the PN classification all functions which differ only by some permutation of the input variables and/or by complementation of one or more of the input variables are considered as being in the same classification entry [3].

where $f_0(x_1,\ldots,x_n)$ and $f_1(x_1,\ldots,x_n)$ are subfunctions of the function $f(x_1,\ldots,x_{n+1})$ for which $x_{n+1}=0$ and $x_{n+1}=1$, respectively.

According to the inductive hypothesis, these subfunctions can be expanded as:

$$f_0(x_1,\ldots,x_n) = (P_1 + P_2 + \ldots + P_p) \oplus (P_{p+1} + P_{p+2} + \ldots + P_m)$$

$$f_1(x_1,\ldots,x_n) = (Q_1 + Q_2 + \ldots + Q_r) \oplus (Q_{r+1} + Q_{r+2} + \ldots + Q_s),$$

with $m \le 5\cdot 2^{n-4}$, $s \le 5\cdot 2^{n-4}$, and for some p and r such that $1 \le p \le m$, $1 \le r \le s$. Then:

$$\begin{aligned}
f(x_1,\ldots,x_{n+1}) &= \overline{x}_{n+1}\ f_0(x_1,\ldots,x_n) + x_{n+1}\ f_1(x_1,\ldots,x_n) && \{\text{Shannon decomposition}\}\\
&= \overline{x}_{n+1}\ ((P_1+\ldots+P_p) \oplus (P_{p+1}+\ldots+P_m)) + x_{n+1}\ ((Q_1+\ldots+Q_r) \oplus \\
&\quad \oplus (Q_{r+1}+\ldots+Q_s)) && \{\text{substitution}\}\\
&= (\overline{x}_{n+1}(P_1+\ldots+P_p) + x_{n+1}(Q_1+\ldots+Q_r)) \oplus (\overline{x}_{n+1}(P_{p+1}+\ldots+P_m)+ \\
&\quad + x_{n+1}(Q_{r+1}+\ldots+Q_s)) && \{\text{Property 1}\}\\
&= (\overline{x}_{n+1}P_1+\ldots+\overline{x}_{n+1}P_p + x_{n+1}Q_1+\ldots+x_{n+1}Q_r) \oplus (\overline{x}_{n+1}P_{p+1}+\ldots \\
&\quad \ldots + \overline{x}_{n+1}P_m + x_{n+1}Q_{r+1}+\ldots+x_{n+1}Q_s) && \{\text{Distributivity of } \cdot \text{ over } +\}
\end{aligned}$$

Since $m \le 5\cdot 2^{n-4}$ and $s \le 5\cdot 2^{n-4}$, therefore $p + r + (m-p) + (s-r) = m + s \le 5\cdot 2^{(n+1)-4}$.

□

Having proved the theorem, we can make a comparison between the three bounds (see Table 1):

1. the upper bound on the number of products in the conventional sum-of-product (AND-OR) expression of logic functions 2^{n-1};
2. the upper bound on the number of products in the AND-XOR expansion in which both complemented and uncomplemented forms of a variable are used $3\cdot 2^{n-3}$;
3. the upper bound on the number of products in the AND-OR-XOR expansion $5\cdot 2^{n-4}$.

It can be seen from the Table 1, that as n increases, the difference between the number of products in the upper bound for AND-OR-XOR expansion and in the bounds for the other expansions grows progressively. For $n = 10$ the bound for AND-OR-XOR expansion has 192 products less than the bound for AND-OR expansion and 64 products less than the bound for AND-XOR expansion.

Table 1. Upper bounds on the number of products for three expansions.

	Number of variables n									
Expansion type	1	2	3	4	5	6	7	8	9	10
AND-OR	1	2	4	8	16	32	64	128	256	512
AND-XOR	1	2	3	6	12	24	48	96	192	384
AND-OR-XOR	1	2	3	5	10	20	40	80	160	320

The following example shows AND-OR, AND-XOR and AND-OR-XOR expansions for a Boolean function of four variables.

Example: Consider the Boolean function of four variables $f(x_1, x_2, x_3, x_4)$ shown in Figure 1.

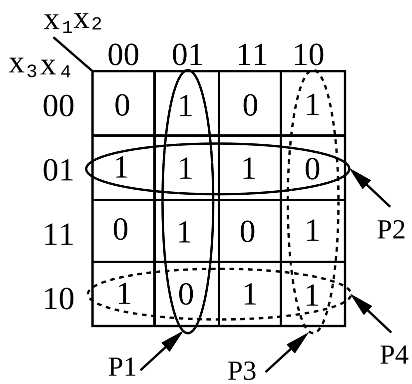

Figure 1: Karnaugh map of the function $f(x_1, x_2, x_3, x_4)$.

The minimal sum-of-products expansion for this function consists of 8 product terms:

$$f(x_1, x_2, x_3, x_4) = \overline{x}_1 x_2 \overline{x}_3 + \overline{x}_1 x_2 x_4 + \overline{x}_1 \overline{x}_3 x_4 + x_2 \overline{x}_3 x_4 + x_1 \overline{x}_2 \overline{x}_4 + x_1 \overline{x}_2 x_3 + x_1 x_3 \overline{x}_4 + \overline{x}_2 x_3 \overline{x}_4$$

The minimal AND-XOR expansion in which both complemented and uncomplemented forms of a variable are used consists of 6 product terms:

$$f(x_1, x_2, x_3, x_4) = x_1 x_3 \oplus x_2 x_4 \oplus \overline{x}_1 \overline{x}_3 \overline{x}_4 \oplus \overline{x}_1 \overline{x}_2 \overline{x}_3 \oplus \overline{x}_2 \overline{x}_3 \overline{x}_4 \oplus \overline{x}_1 \overline{x}_2 \overline{x}_4$$

The minimal AND-OR-XOR expansion consists of 4 product terms:

$$f(x_1, x_2, x_3, x_4) = (\overline{x}_1 x_2 + \overline{x}_3 x_4) \oplus (x_1 \overline{x}_2 + x_3 \overline{x}_4) = (P_1 + P_2) \oplus (P_3 + P_4)$$

The minterms, corresponding to the product terms P_1, P_2, P_3, P_4 are shown on Figure 1.

3 Conclusion

The representation of logic functions as an XOR sum of two sum-of-products expressions is considered. The upper bound on the number of products in such the representation is shown to be $5 \cdot 2^{n-4}$, where n is the number of variables. This bound is 37.50% smaller than the upper bound on the number of products in the conventional sum-of-product expression of logic functions (2^{n-1}), and 16.67% smaller than the upper bound on the number of products in the AND-XOR expansion in which both complemented and uncomplemented forms of a variable are used ($3 \cdot 2^{n-3}$). In addition, since only one XOR operation per function is used, an AND-OR-XOR expansion could have an advantage over an AND-XOR expansion for technologies like CMOS, in which an XOR operation has a more complex implementation than an OR operation.

While the proposed expansion leads to a three-level implementation which is good for achieving the area gains over two-level implementations, on the other hand finding such an expansion may be more problematic than finding AND-OR and AND-XOR expansions. So, some powerful algorithm has to be found to eliminate this problem.

References

[1] D.E. Muller, *Application of Boolean algebra to switching circuit design and to error detection,* IRE Trans. Electron. Comput., vol. EC-3, September 1954, pp. 6-12.

[2] I.S. Reed, *A class of multiple-error-correcting codes and their decoding scheme,* IRE Trans. on Inform. Theory, vol. 4, September 1954, pp. 38-42.

[3] S.L. Hurst, D.M. Miller, J.C. Muzio, *Spectral Techniques in Digital Logic,* Academic Press Inc., 1985.

[4] T. Sasao, P. Besslich, *On the complexity of Mod-2 Sum PLA's,* IEEE Trans. on Computers, Vol. 39, No. 2, February 1990, pp. 262-266.