TOP: An Algorithm for Three-Level Optimization of PLDs

E. Dubrova P. Ellervee
Department of Electronics
Royal Institute of Technology
S-164 40 Kista, Sweden
{elena, lrv}@ele.kth.se

D. M. Miller J. C. Muzio
Department of Computer Science
University of Victoria
Victoria, B.C., Canada, V8W 3P6
{mmiller, jmuzio}@csr.uvic.ca

In this paper we present an heuristic algorithm TOP (**T**hree-level **O**ptimization of **P**LDs), targeting a three-level logic expression of type $g_1 \circ g_2$, where g_1 and g_2 are sum-of-products and "$\circ$" is a binary operation. Such an expression can be implemented by a three-level Programmable Logic Device (PLD) consisting of PLA1 and PLA2, implementing the first two levels of logic, and a set of two-input *logic expanders*, implementing the third level. Each logic expander can be programmed to realize any function of two variables. PLD of this type seems to give a good trade-off between the speed of a flat PLA and density of a multi-level network of PLAs. TOP chooses the functionality of the logic expanders so that the area of the PLAs is minimized. To the best of our knowledge, this is the first work addressing this problem for an arbitrary operation "$\circ$" and attempting to choose the operation which results in the smallest total number of product-terms. Several algorithms for the specified cases of "$\circ$" have been presented in the past (see [2] for overview). An algorithm, constructing the expansion of type $f(X) = g(X_g) \circ h(X_h)$ for an arbitrary "$\circ$" with $X_g, X_h \subset X$ and $X_g \cup X_h = X$ is described in [3]. However, this algorithm does not target the minimal number of products, and does not consider the case when X_g or X_h equals X, which is allowed in our case.

TOP works as follows. Given a Boolean function $f = (F_f, D_f, R_f)$, where F_f, R_f and D_f denote on-set, off-set and don't-care-set of f, respectively, we first decide which operation "$\circ$" to use, and then search for two sets of cubes g_1 and g_2, such that $g_1 \circ g_2 \supseteq F_f$, $(g_1 \circ g_2) \cap R_f = \emptyset$, and the total number of cubes in g_1 and g_2 is minimized.

Making the right choice of "$\circ$" is crucial for the size of the resulting expression. To decide the type of operation, we first run a very simple and fast AND-OR-XOR minimization heuristic where most of the choices are made randomly. If the number of product-terms in the resulting AND-OR-XOR expression is smaller than one in the the initial sum-of-product form, then "$\circ$" is assigned to XOR. Otherwise, we apply a simple AND-OR-AND minimization heuristic. If it improves the cost, then "$\circ$"= AND. Otherwise "$\circ$" is assigned to OR.

For AND-OR-XOR minimization we use the algorithm presented in [2]. AND-OR-AND minimization is performed by the algorithm described in [1]. For AND-OR-OR minimization, we have designed a dynamic greedy constructive algorithm, performing a bipartition of the on-set F_f of f. Clearly, it cannot reduce the number of product-terms in the expression, but it can allow re-distribution of the product-terms in a way favorable for the total area of the targeted PLD. The goal is to divide the cubes between two partitions in such a way that the number of common inputs and outputs for both partitions is minimal.

After "$\circ$" is selected to be XOR, OR or AND and the corresponding three-level expression is found, output phase optimization is applied. In the final implementation, g_1 is implemented by PLA1 and g_2 is implemented by PLA2. The logic expanders are programmed depending on the choice of "$\circ$" and the phase assignment of g_1 and g_2.

The experimental results show that, on average, the total number of cubes in g_1 and g_2 is 3 times smaller than the number of cubes in the two-level AND-OR cover generated by Espresso.

References

[1] E. Dubrova, P. Ellervee "A fast algorithm for three-level logic optimization", *Proc. Int. Workshop on Logic Synthesis* (1999), 251-254.

[2] E. V. Dubrova, et. al., "AOXMIN-MV: A heuristic algorithm for AND-OR-XOR minimization", *Proc. 4th Int. Workshop on Applications of Reed-Muller Expansion in Circuit Design* (1999), 37-53.

[3] T. Stanion, C. Sechen, "Quasi-algebraic decomposition of switching functions", *Proc. Int. Conf. Advanced Research in VLSI* (1995), 358-367.