A PARTITIONING METHOD FOR AREA OPTIMIZATION BY TREE ANALYSIS

Yuichi Nakamura
Kazutoshi Wakabayashi
Tomoyuki Fujita

C&C Systems Research Laboratories
NEC Corporation
4-1-1 Miyazaki Miyamae-ku
Kawasaki 216 JAPAN

ABSTRACT

This paper presents a new method for area optimization method of large scale combinational circuits. The proposed technique partitions a given circuit and then performs collapsing and optimization on each partition. The algorithm selects sub-circuits that are appropriate for collapsing based on tree structure analysis, which requires time that is linearly proportional to the circuit size. Experimental results on ISCAS benchmarks show that this method can achieve up to a 27% reduction in the number of literals compared to the well known simple elimination approach.

6.1 INTRODUCTION

Over the past decade, many techniques for technology independent area optimization for combinational logic circuits have been proposed [1, 3]. Most of the methods adopt a common strategy. First, the entire circuit is flattened into a two-level logic representation. Next, two-level logic minimization is applied to the flattened circuit. Finally, algebraic/Boolean decomposition is performed. The resultant circuit is expressed as optimized multi-level Boolean logic equations [1]. More recently, methods have been proposed that apply multi-level optimizations after decomposition through the use of internal “don’t cares”. When a given circuit is small enough, these methods are able to successfully flatten the entire circuit into two-level logic. However, for large scale circuits involving more than a thousand gates, such flattening and two-level minimization cannot be applied because the computational time and memory space needed
are exponentially proportional to the size of circuit.

Other optimization approaches are based on internal "don't cares" and use Binary Decision Diagrams [2] or test pattern generation [7]. These methods can handle relatively large scale circuits without flattening the entire circuit [3, 4]. However, they still cannot effectively deal with large scale circuits involving more than several thousand gates or circuits with certain structures such as multipliers. For example, it has been proved that it requires exponential computation time to construct the BDD for a multiplier [2].

This paper introduces a new efficient partitioning technique, based on tree analysis, to optimize circuit area by partitioning a circuit and partially flattening it to a two-level representation. A Boolean network [1] is constructed for the circuit, and trees from each output of the network are isolated. These network trees are then analyzed to obtain co-trees of the network. This information is then used in partitioning the original circuit into sub-circuits to which partial collapsing and logic optimization are applied. Details of this new approach are discussed in the following sections, and experimental results from the ISCAS benchmarks are presented.

6.2 LOGIC PARTITION AND PARTIAL COLLAPSING

6.2.1 Collapsing for Optimization

Collapsing is a well known technique and has been frequently used for circuit minimization. When a multi-level circuit is small enough, it is common to flatten the circuit to a two-level one first. However, since flattening the entire circuit requires listing all of its min-terms, it is not always feasible to do so. Even though the entire circuit cannot be flattened, collapsing can still be performed on selected portions of the circuit, and this has been shown to be sufficiently effective for area optimization.

For example, Fig. 6.1 shows a multi-level circuit, which consists of 19 literals. By focusing on the multiple fanout point in this circuit, it is possible to partition the circuit into two "reconvergent" parts. After applying the collapsing process to each part instead of flattening to two-level entirely, the literal number increases to 21 (Fig. 6.2) compared with initial circuit. The circuit can then be