Efficient FPGA Mapping using Priority Cuts

Sungmin Cho **Satrajit Chatterjee** **Alan Mishchenko** **Robert Brayton**

Department of EECS, University of California, Berkeley
{smcho, satrajit, alanmi, brayton}@eecs.berkeley.edu

Categories and Subject Descriptors
B.6.3 [**Logic Design**]: Design Aids—*Optimization*; B.7.1 [**Integrated Circuits**]: Types and Design Styles—*Gate arrays*; J.6 [**Computer-Aided Engineering**]: *Computer-aided design (CAD)*

General Terms
Algorithms

Keywords
FPGA, Technology Mapping, Cut Enumeration, Area Recovery, Lossless Synthesis

Abstract

The poster presents a new algorithm for FPGA technology mapping into K-input look-up tables (LUTs) applicable to large combinational and sequential circuits.

The proposed algorithm is similar to the traditional structural FPGA mapping algorithms which perform depth-oriented LUT mapping followed by area recovery. The traditional methods enumerate all or nearly all cuts of each node in order to find an optimum-depth mapping. The proposed algorithm avoids exhaustive cut enumeration by computing and updating only a small number (typically, 5-10) of “good” K-feasible cuts at each node. These cuts are called “priority cuts”. The criteria used to prioritize the cuts differ depending on the mapping goals. For example, when mapping for delay, the cuts are prioritized first by delay, then by size, and finally by area flow. The experiments indicate that such prioritization leads to a depth-optimal mapping for 95% of all benchmarks with any LUT sizes, even if only one cut is stored at each node. Increasing the number of priority cuts to 8 allows the algorithm to avoid area penatly due to not enumerating all cuts, while offering dramatic improvements in memory and runtime.

For 6-input LUTs with 8 cuts stored at each node, both memory and runtime are reduced about 5x, compared to state-of-the-art mapping, while delay and area are comparable. For 8-input and larger LUTs, the improvements in memory and runtime often reach 100x.

The runtime and memory requirements of the proposed algorithm are linear in the number of nodes in the subject graph and in the cut size. This makes it useful for mapping into large macro-cells, which can implement a subset of Boolean functions with a given number of inputs. Another promising extension is a sequential mapping, which leads to a 20% reduction in delay, compared to the combinational mapping and retiming performed as a postprocessing step.