Towards More Efficient Logic Blocks By Exploiting Biconditional Expansion

Pierre-Emmanuel Gaillardon, *EPFL*
Gain Kim, *EPFL*
Xifan Tang, *EPFL*
Luca Amarù, *EPFL*
Giovanni De Micheli, *EPFL*

Nowadays, *Field Programmable Gate Arrays* (FPGA) exploit *Look-Up Tables* (LUTs) to generate logic functions. A *K*-input LUT can implement any Boolean functions with *K* inputs. Thanks to this flexibility, LUTs remained conceptually unchanged in FPGAs, only the number of inputs increased in time. Unfortunately, the flexibility does not come for free and LUTs have non-negligible costs in both circuit-level performances (large number of memories, area or delay penalties) and logic-level capabilities (limited fan-out). Here, we propose an FPGA fabric based on two novel logic blocks. First, we introduce a new LUT design showing reduced power consumption with no sacrifice in the logic flexibility. Then, we present a block suited to arithmetic functions but preserving enough versatility to implement general logic functions. The two blocks are supported by a recently introduced logic representation called *Biconditional Binary Decision Diagrams* (BBDDs). Using architectural-level benchmarking, we showed that an FPGA architecture exploiting the novel blocks performs significantly better than current state-of-the-art FPGA architectures at 40nm technological node over a large set of test circuits. While reducing the power consumption of MCNC big20 benchmarks by 29%, the proposed architecture is able to efficiently implement arithmetic circuits as compared to its traditional LUT-based FPGA counterpart. For instance, a 256-bit adder can be realized with a 43% gain in area×delay product. While considering large general and arithmetic logic benchmarks, we observe, on average, 4%, 3% and 10% improvements in area, delay and power respectively.

ACM Categories & Descriptors: H.3.6 Programmable logic elements

Keywords: FPGA; Arithmetic functions; BBDD; Logic Element

DOI: http://dx.doi.org/10.1145/2684746.2689100