

Preface

Realization of high performance arithmetic circuits targeted towards a specific family of the high-end Field Programmable Gate Arrays (FPGAs) continue to remain a challenging problem. Many fast arithmetic circuits proposed over the decades may not be amenable to efficient realization on a selected FPGA architecture. Experience has shown that current CAD tools for FPGAs are often unable to infer the native architectural components efficiently from the given input Hardware Description Language (HDL) specification of the circuit, as they explore only a small design space close to the input architectural description. The logic synthesis techniques inherent to the CAD tools are also often unable to apply the proper Boolean identities and perform appropriate algebraic factoring and sub-expression sharing, especially when intermediate signals are tapped out or registered. *Primitive instantiation* is an effective approach for optimization of designs on the Xilinx FPGA platform, and is often simpler than rewriting the Register Transfer Level (RTL) code to coax the logic synthesis tool to infer the desired architectural components. In addition, the FPGA CAD tools often fail to achieve an efficient placement of logic blocks on the FPGA fabric, resulting in higher routing delays.

In this book, we describe the optimized implementations of several arithmetic datapath, controlpath, and pseudorandom sequence generator circuits. We explore regular, modular, cascadable, and bit-sliced architectures for these circuits, by directly instantiating the target FPGA-specific primitives in the HDL specifications of the circuits. We justify every proposed architecture with detailed mathematical analyses. We improve performance by enforcing a constrained placement of the circuit building blocks, by placing the logically related hardware primitives in close proximity to one another, thereby minimizing the routing delay. This is accomplished by supplying relevant placement constraints in the Xilinx proprietary “User Constraints File” (.ucf) format to the FPGA CAD tool.

Taking advantage of the regularity of the architectures of the circuits proposed by us, the HDL specifications of the circuits as well as the placement constraints can be automatically generated. We have implemented a GUI-based CAD tool named *FlexiCore* integrated with the *Xilinx ISE* (Integrated Software Environment)

design environment for design automation of platform-specific high performance arithmetic circuits from user-level specifications. This tool was used to implement the proposed circuits, as well as hardware implementations of two integer arithmetic algorithms (Greatest Common Divisor (GCD) using Binary GCD algorithm and matrix multiplication using Distributed Arithmetic (DA)) where several of the proposed circuits were used as building blocks. Implementation results demonstrate higher performance and superior operand-width scalability at acceptable power-delay product (PDP) for the proposed circuits, with respect to implementations derived through other existing approaches.

Kharagpur

Ayan Palchaudhuri
Rajat Subhra Chakraborty

High Performance Integer Arithmetic Circuit Design on
FPGA

Architecture, Implementation and Design Automation

Palchaudhuri, A.; Chakraborty, R.S.

2016, XVII, 114 p. 56 illus., Hardcover

ISBN: 978-81-322-2519-5