Chapter 11
An Educational Tool for Digital Electronic System Synthesis and Optimization

Hakduran Koc
University of Houston – Clear Lake, USA

Seyit Ozturk
University of Houston – Clear Lake, USA

ABSTRACT

Considering the complexity of today's digital electronic systems, it is crucial to have open-source electronic design automation (EDA) tools specifically developed for educational purposes. Such tools can easily be modified to meet the demands of the course being taught and they can be configured to expose the intermediate steps during the design process. This chapter presents an educational EDA tool to help students better understand and implement fundamental concepts in digital electronic design and synthesis courses. The tool receives an intermediate format that represents the target system behavior and a set of constraints as input, and generates the representation of the actual circuit using high-level electronic components such as functional units, memory, and steering logic components available in its technology library. It considers execution delay, area, memory space consumption, and reliability constraints. The user is able to interact with the tool during the design process and select the algorithms to perform various synthesis and optimization tasks.

DOI: 10.4018/978-1-5225-5673-2.ch011

INTRODUCTION

Electronics systems are indispensable part of our daily life; ranging from simple circuits in kid toys to smart phones to mission-critical avionics systems that manage hundreds of sub-systems to perform complex tasks. Digital electronic systems combine particular logic and circuit design techniques required to implement Integrated Circuits (ICs). Such circuits consist of miniaturized electronic components built into an electrical network on a semiconductor board. Digital IC design produces components such as microprocessors, Field Programmable Gate Arrays (FPGAs), memories (RAM, ROM, and flash), Application Specific Integrated Circuits (ASICs) (Coussy, & Takach, 2009). Digital system design puts emphasis on maximizing performance, reducing power/energy consumption, minimizing memory space consumption, improving reliability, verifying logical/functional correctness, and maximizing circuit density.

Digital system synthesis is a process in which an abstract form of desired circuit behavior is turned into a design implementation in terms of the components available in a technology library at a given abstraction level (Makris, & Orailoglu, 1999). Circuit behavior is typically given using a Hardware Description Language (HDL) such as VHDL, Verilog, or SystemC (Sun, 1994). After analyzing the behavioral source code, it is translated into an intermediate format such as netlist, state diagram, dataflow and sequencing graphs. Various Electronic Design Automation (EDA) tools use this intermediate format as input in order to generate the final circuit that meets all design constraints (Martin, & Smith, 2009). Among others, performance, area, power/energy consumption, reliability, and memory efficiency are important design optimization metrics. An EDA tool not only has to meet all design constraints but also aims at improving design metrics as much as possible.

Raising the abstraction level (hiding unnecessary details) in the synthesis process brings many advantages including reduced design time, less probability of design errors, and reduced complexity (Sarkar, S., Dabral, S., Tiwari, PK., & Mitra, RS., 2009). Due to these advantages, High Level Synthesis (HLS) has become increasingly popular in EDA field. In addition, HLS tools can reduce the verification time (which is a major contributor in overall design cycle) and can optimize the final circuit and create opportunities for extensive design space exploration (Duranton, M., Yehia, S., De Sutter, B., De Bosschere, K., Cohen, A., Falsafi, B., Gaydadjiev, G., Katevenis, M., Maebe, J., Munk, H., Navarro, N., Ramirez, A., Temam, O., & Valero, M., 2009). The followings are examples of electronic components utilized to generate the final circuit in HLS: ALU and multiplier as functional units, registers as memory components, and multiplexer and bus as steering logic components. Three main tasks in High Level Synthesis are resource allocation, operation scheduling, and resource sharing/binding. In resource allocation phase, the number and type of resources that