Chapter 1 Introduction

This module is designed to focus on electronic system design and analysis through simulation and implementation adopting industry standard Electronic Design Automation methodologies. Students will develop know-how in digital design, system partitioning, design verification and implementation. In analog design and simulation, analysis techniques such as DC operating point, DC sweep, transient and AC analyses will also be covered. This module provides the students with circuit design and analysis skillsets to support the needs of the electronics or related industry. The prerequisite to take this module is that students should have successfully completed the year one modules on Electrical Principles, Electric Circuits, Analog Electronics and Digital Electronics.

1.1 Overview of Electronic Design Automation

Electronic Design Automation (EDA or ECAD) are computer-based software that are design to speed up the process of creating complex electronic circuits, system, and products such as printed circuit boards and integrated circuits. The tools work together in a design flow that chip designers use to design and analyze entire semiconductor chips. EDA tools cover a broad area of technologies including Design Entry, Synthesis, Digital and Analog Simulation, Programmable Logic Devices (PLD) design and Application Specific Integrated Circuits (ASIC) design.

Early Days

Before EDA, integrated circuits were designed by hand, and manually laid out. Some advanced shops used geometric software to generate the tapes for the Gerber photo plotter. The process was fundamentally graphic, with the translation from electronics to graphics done manually. The best known company from this era was Calma, whose GDSII format survives.

By the mid-1970s, developers started to automate the design along with the drafting. The first placement and routing (Place and Route) tools were developed.

The next era began about the time of the publication of "Introduction to VLSI Systems" by Carver Mead and Lynn Conway in 1980. This ground breaking text advocated chip design with programming languages that compiled to silicon. The immediate result was a considerable increase in the complexity of the chips that could be designed, with improved access to design verification tools that used logic simulation. Often the chips were easier to lay out and more likely to function correctly, since their designs could be simulated more thoroughly prior to construction. Although the languages and tools have evolved, this general approach of specifying the desired behavior in a textual programming language and letting the tools derive the detailed physical design remains the basis of digital IC design today.

Birth of Commercial EDA

1981 marks the beginning of EDA as an industry. For many years, the larger electronic companies, such as Hewlett Packard, Tektronix, and Intel, had pursued EDA internally. In 1981, managers and developers spun out of these companies to concentrate on EDA as a business. Daisy Systems, Mentor Graphics and Valid Logic Systems were all founded around this time, and collectively referred to as **DMV**. Within a few years there were many companies specializing in EDA, each with a slightly different emphasis. The first trade show for EDA was held at the Design Automation Conference in 1984.

In 1981, the U.S. Department of Defense began funding of VHDL as a hardware description language. In 1986, Verilog, another popular high-level design language, was first introduced as a hardware description language by Gateway Design Automation. Simulators quickly followed these introductions, permitting direct simulation of chip designs: executable specifications. In a few more years, back-ends were developed to perform logic synthesis.

Current Status

Current digital flows are extremely modular. The front ends produce standardized design descriptions that compile into invocations of "cells", without regard to the cell technology. Cells implement logic or other electronic functions using a particular integrated circuit technology. Fabricators generally provide libraries of components for their production processes, with simulation models that fit standard simulation tools. Analog EDA tools are far less modular, since many more functions are required, they interact more strongly, and the components are (in general) less ideal.

EDA for electronics has rapidly increased in importance with the continuous scaling of semiconductor technology. Some users are foundry operators, who operate the semiconductor fabrication facilities, or "fabs", and design-service companies who use EDA software to evaluate an incoming design for manufacturing readiness. EDA tools are also used for programming design functionality into FPGAs.

1.2 Digital Design Flow

Typical Digital FPGA Design Flow

EDA software makes it easy to implement a desired logic circuit by using a programmable logic device, such as a Field Programmable Gate Array (FPGA) chip. A typical FPGA design flow is illustrated in Figure 1.1.

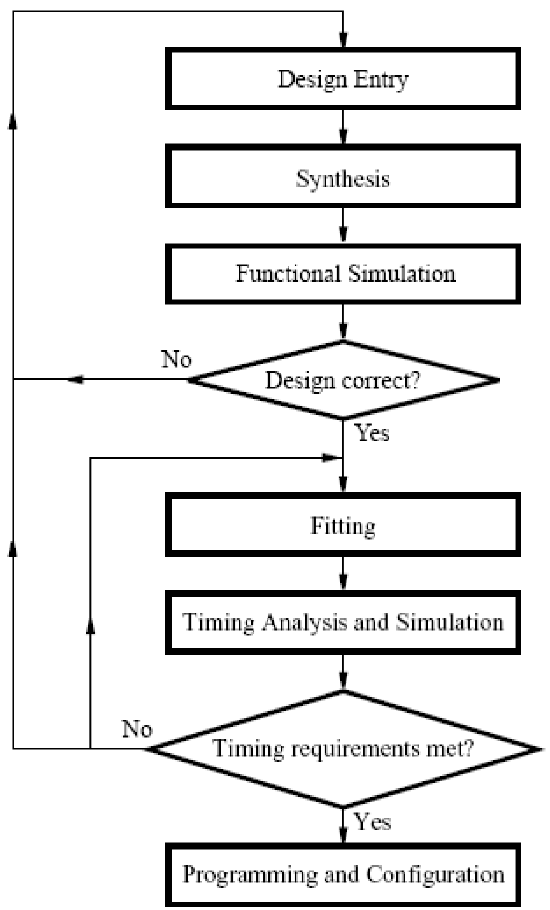

Figure 1.1 Typical FPGA design flow

The FPGA design flow involves the following steps:

• Design Entry – the desired circuit is specified either by means of a schematic diagram, or by using a hardware description language, such as Verilog or Very-High-Speed Integrated Circuit Hardware Description Language (VHDL).

• Synthesis – the entered design is synthesized into a circuit that consists of the logic elements (LEs) provided in the FPGA chip.

• Functional Simulation – the synthesized circuit is tested to verify its functional correctness; this simulation does not take into account any timing issues.

• Fitting – the ECAD Fitter tool determines the placement of the logic elements (LEs) defined in the netlist in an actual FPGA chip; it also chooses routing wires in the chip to make the required connections between specific LEs.

• Timing Analysis – propagation delays along the various paths in the fitted circuit are analyzed to provide an indication of the expected performance of the circuit.

• Timing Simulation – the fitted circuit is tested to verify both its functional correctness and timing.

• Programming and Configuration – the designed circuit is implemented in a physical FPGA chip by programming the configuration switches that configure the LEs and establish the required wiring connections.

1.3 Analog Design Flow

The industry trend is to replace analog circuits with digital wherever possible, but there are still some functions where analog parts cannot be avoided. The most obvious are applications in communications, filters, amplifiers and RF circuits. Analog circuits are important and necessary in a large range of applications and will remain an important component in many integrated circuits.

The Analog Design Flow

The design of digital systems is regarded as quite a clean flow; and automation in both front-end and layout, including verification is in common use. The analog world lacks much more automation. Analog synthesis is difficult. One major reason for this non-satisfactory situation is that there are many kinds of analog parameters to be treated in design and to be verified in test benches, whereas in digital circuits there are fewer parameters (like speed, power and chip area).

In Figure 1.2 a flow graph for traditional manual cell-level analog circuit design is shown. The circuit specification puts lower and/or upper limits on some performance metrics while other may be less precisely specified.

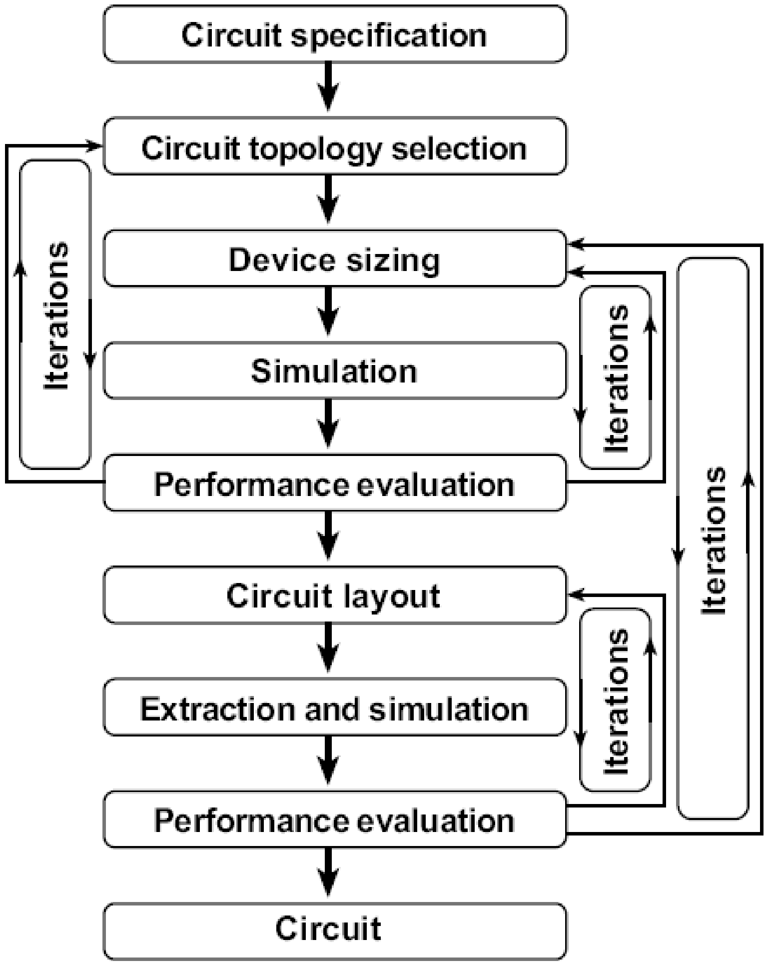

Figure 1.2 Flow graph of the cell-level manual analog circuit design

1.4 Design Flow Integration

One of today key challenges in electronic design is the ability to integrate the various design flows. Figure 1.3 shows an integrated flow for digital and analog (mixed-signal) design while Figure 1.4 shows an integrated design flow for hardware and software (firmware) design.

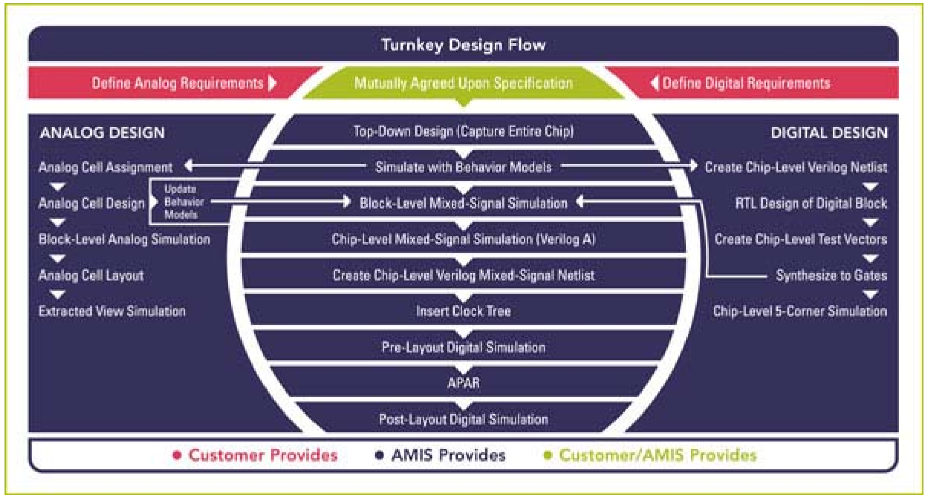

Figure 1.3 Integrated digital and analog (mixed-signal) design flow

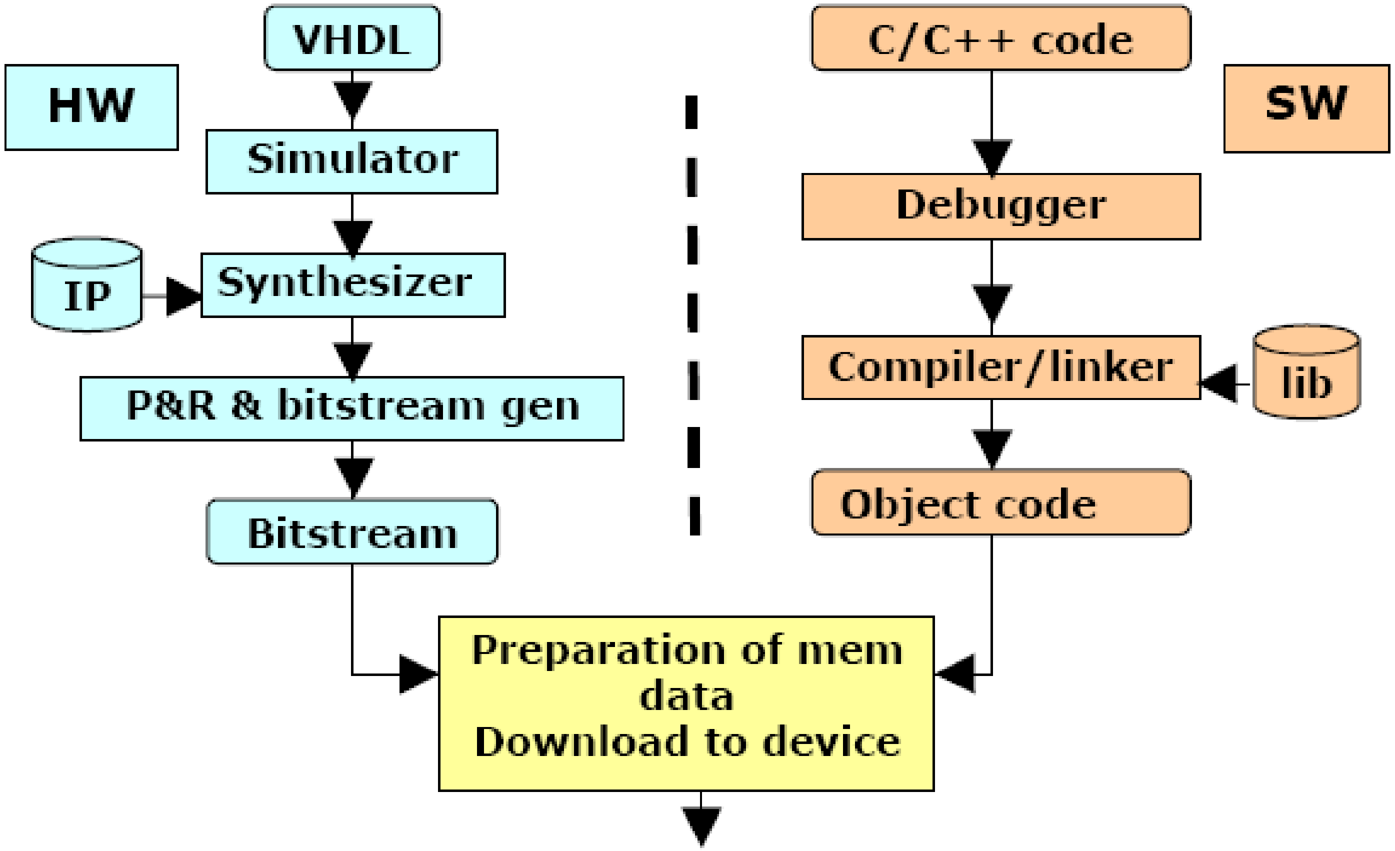

Figure 1.4 Integrated hardware and software (firmware) design flow