

CS 220: Synthesis of Digital Systems

(Currently listed as “Silicon Compilation”)

Spring, 1998

University of California, Riverside

Course information

Instructor	Frank Vahid (vahid@cs.ucr.edu). Office: Bourns A207
Lecture	TR 8:10-9:30, OLMH 2107
Textbooks	Hsu, <i>VHDL Modeling for Digital Design Synthesis</i> , Kluwer Academic Publishers, ISBN 0-7923-9597-2.
Prerequisite	Courses in programming and digital design
Course call # and units	16119, 4 units.

Course objective

To learn the theory and practice of the automated synthesis of digital systems.

Course grade

Labs	40%
Homeworks	20%
Exams	40%

Course overview

The design of digital hardware has been revolutionized in the past several years. Rather than connecting gates, designers instead create a model describing desired functionality using a program-like language, simulate the model to ensure it correctly and completely captures the intended functionality, and then automatically synthesize the model into gates. Thus, synthesis in many ways fills the same role as compilation in the software world. The demand for designers who can comfortably design both hardware and software is very high, especially since most embedded systems today (e.g., cell phones, pagers, automotive electronics, etc.) consist of both hardware and software on a single chip, but availability of such designers is low, enabling such designers to “name their price” in the words of several industry managers.

This course will focus on synthesis of digital systems. We will model systems in VHDL, learn to write VHDL that synthesizes into good hardware, and study the underlying synthesis principles and algorithms. We will also analyze the IP revolution: because of increasing chip capacities, companies sell VHDL/Verilog description of components intended to reside on chip with other components, rather than selling physical packages. This revolution has opened the design field to large numbers of small startup companies that can sell their “product” (VHDL synthesizable descriptions) without incurring the high-cost of generating actual chips.

Course contents

1. VHDL – top-down modeling, simulation, testbenches
2. Logic synthesis
3. FSM synthesis
4. FSMD synthesis (Register-transfer synthesis)
5. Synthesis from high-level programs (behavioral synthesis)
6. Synthesis for FPGA's
7. Intellectual Property and cores