# Intermediate VHDL Coding Styles for Synthesis

**2 Days: 50% lecture, 50 % Lab** **Intermediate Level**

## Overview

An in-depth study of VHDL coding styles, methodologies, and design techniques used to efficiently synthesize digital hardware (ASICs and FPGAs).

Class topics focus on mapping digital hardware structures to vendor independent VHDL code. Detailed do's and don'ts of synthesis coding styles are discussed. Lecture and laboratory materials illustrate the optimization differences achieved by different VHDL coding styles. Students will learn proven coding practices that result in smaller and faster designs.

Synthesis coding styles recommended in this course are also recommended by IEEE standard 1076.6, Standard For VHDL Register Transfer Level Synthesis. As a result, no matter whose synthesis tool you use, the synthesis coding styles and techniques you learn will yield effective results.

This class is the first half of VHDL Coding for Synthesis.

## Intended Audience

The numerous examples in this course make it suitable for a student with limited VHDL. The application focus of this course results in the student being ready for VHDL based ASIC or FPGA design.

## Course Objective

Upon completion of this course, students will be able to:

- Write efficient, vendor independent VHDL code
- Use best coding styles and practices to create ASIC and FPGA logic
- Understand and avoid problematic hardware coding styles
- Use code templates to create a design from the block diagram
- Read VHDL code and draw the corresponding hardware (reverse engineer)
- Use types, overloading, and conversion functions from standard VHDL packages (std_logic_1164, numeric_std, and std_logic_arith)

# Intermediate VHDL Coding Styles for Synthesis

## Course Outline

**Day 1, Module Syn1**

- Synthesis Overview
- Combinational Logic
- Registers and Latches
- UART Transmitter:
  - RTL Code + Statemachine

**Day 2, Module Syn2**

- Numeric Types and Packages
- Arithmetic Logic
- Comparison and Multiplication
- Partitioning
- Synthesis Process

## Prerequisites

Students taking this course should have working knowledge of digital circuits and prior exposure to VHDL through experience or the course:

**Comprehensive VHDL Introduction - 4 days**

## Follow-On Courses

Students wishing to go beyond what they learned in this course should take:

**Advanced VHDL Coding for Synthesis - 2 days**

## Other Recommended Courses

Students may also be interested in the following companion course:

**VHDL Testbenches and Verification – 4 days**

All of our courses can be customized to meet your specific needs. Either see our website or contact us for details.

## Customization

All of our courses can be customized to meet your specific needs. Either see our website or contact us for details.

## Training Approach

This hands-on, how-to course is taught by experienced hardware designers using a computer driven projector. We prefer and encourage student and instructor interaction. Questions are welcome. Bring problematic code.

## Contact

To schedule a class or for more information, contact:

Jim Lewis
Director of Training
(800) 505-8435 / (800) 505-VHDL
jim@SynthWorks.com
http://www.SynthWorks.com

**Learn VHDL from a designer's perspective with SynthWorks.**