Chapter 4

Synthesis of Sequential Circuits

Chapter 3 covered the basic synthesis rules for clocked processes and the fact that they could be used to design sequential logic functions. This chapter presents the first discussions of how to design common sequential circuit functions using VHDL synthesis, and uses various types of counters as the vehicle to illustrate the techniques. Design for testability will also be addressed as it applies to general counting applications.

Counters are the basic building blocks of a number of different kinds of sequential logic functions, and because of their importance will be discussed in detail in this section. Design techniques will be developed gradually through a series of examples, beginning with the basic counter syntax and proceeding to add the capabilities normally needed in typical
real world systems. Following the general discussion of counters, examples of how they can be synthesized into other types of applications will be demonstrated in chapter 5.

It will become clear in the next few chapters how synthesis can help create good logic designs while still giving the designer control over the basic design decisions and techniques used. In fact, it will be seen that designer experience is essential to the synthesis process in most cases and allows design work to move to a higher level while using the synthesis tool to handle the gate level details.

**Basics of counters**

In the current industry practice there are many different ways to implement counters, such as those that count up, down, or either up and down depending on a control signal. In most applications, counters have separate control signals to enable counting as well as to reset or load the counter to a desired state. Although binary counters are the most common form of counter design, other variations such as BCD (Binary Coded Decimal), Johnson, Gray or pseudo-random counters may be preferred depending on the application. Fortunately all of these counter types are readily synthesizable from VHDL constructs.

As explained in chapter 3, for any particular synthesis tool various VHDL packages are usually provided by the tool vendor that handle operations such as type conversions, operator overloading, and synthesis specific attributes and which make the VHDL coding task simpler and easier to understand. These capabilities will be pointed out when they are first used in the examples below in order to clarify their usage. In addition, although synthesizers can often handle various VHDL signal types such as Bit and Std_Logic, this book will use Std_Ulogic most often since this IEEE-adopted 9 state logic definition yields better flexibility during VHDL simulation.