This page intentionally left blank.

CHAPTER 10
Logic Design and Finite State Machines

A large component of digital systems design is the implementation of application-specific algorithms. In some cases, software implements the bulk of algorithms, because microprocessors are flexible and easily programmable for a wide range of functions. Such systems may use only hardware to acquire and store data on behalf of the software. Other systems may be unable to perform their intended tasks with software alone. The reasons for this vary by application and often include throughput requirements that are not practically achievable with a microprocessor.

When algorithms are implemented in hardware, state machines are often employed to accomplish the task. A state machine can be made arbitrarily complex and can function similarly to software running on a microprocessor. Just as software moves through a sequence of tiny steps to solve a larger problem, a state machine can be designed to advance when certain conditions are satisfied. As the state machine progresses, it can activate other functions, just as software requests transactions from a microprocessor's peripherals.

This chapter focuses on higher-level logic design techniques used to implement functions ranging from basic microprocessor address decoding to clock domain crossing to state machines. Few design methodologies are mandatory, but certain techniques make the design task easier by freeing an engineer from having to worry about mundane low-level details while concentrating on the higher-level functions. Most software today is not written in assembly language, because the high-level language methodology is so productive. A software engineer does not have to worry about the accumulator and addressing modes of a microprocessor and can concentrate on the specific application being written. Likewise, more automated logic design techniques such as *hardware description languages* (HDL) handle the tasks of logic minimization and gate interconnection so that a hardware engineer can spend more time implementing the necessary algorithms for a specific application.

A great thing for the design community is that HDLs, once used by a relatively small set of companies with hefty financial resources, are becoming less expensive and more accessible with each passing year. The investment required to begin with these tools at the entry level ranges from nearly free up to a few thousand dollars, depending on the features desired and the subsidization that may be available. Because of the wide popularity of HDLs, several companies that manufacture programmable logic devices provide free or low-cost development tools to their customers. Stand-alone entry-level HDL tools are also available from companies including Model Technology and Simucad.

10.1 HARDWARE DESCRIPTION LANGUAGES

Many basic peripheral logic functions are available in off-the-shelf ICs. A variety of UART ICs are available, DMA controllers are available, and simple address decoding can be accomplished with