1.5 Project

Objective Implement all the logic gates presented in the chapter. The only building blocks that you can use are primitive Nand gates and the composite gates that you will gradually build on top of them.

Resources The only tool that you need for this project is the hardware simulator supplied with the book. All the chips should be implemented in the HDL language specified in appendix A. For each one of the chips mentioned in the chapter, we provide a skeletal .hdl program (text file) with a missing implementation part. In addition, for each chip we provide a .tst script file that tells the hardware simulator how to test it, along with the correct output file that this script should generate, called .cmp or "compare file." Your job is to complete the missing implementation parts of the supplied .hdl programs.

Contract When loaded into the hardware simulator, your chip design (modified .hdl program), tested on the supplied .tst file, should produce the outputs listed in the supplied .cmp file. If that is not the case, the simulator will let you know.

Tips The Nand gate is considered primitive and thus there is no need to build it: Whenever you use Nand in one of your HDL programs, the simulator will automatically invoke its built-in tools/builtIn/Nand.hdl implementation. We recommend implementing the other gates in this project in the order in which they appear in the chapter. However, since the builtIn directory features working versions of all the chips described in the book, you can always use these chips without defining them first: The simulator will automatically use their built-in versions.

For example, consider the skeletal Mux.hdl program supplied in this project. Suppose that for one reason or another you did not complete this program's implementation, but you still want to use Mux gates as internal parts in other chip designs. This is not a problem, thanks to the following convention. If our simulator fails to find a Mux.hdl file in the current directory, it automatically invokes a built-in Mux implementation, pre-supplied with the simulator's software. This built-in implementation—a Java class stored in the built In directory—has the same interface and functionality as those of the Mux gate described in the book. Thus, if you want the simulator to ignore one or more of your chip implementations, simply move the corresponding .hdl files out of the current directory.

Steps We recommend proceeding in the following order:

0. The *hardware simulator* needed for this project is available in the tools directory of the book's software suite.

1. Read appendix A, sections A1-A6 only.

2. Go *through the hardware simulator tutorial,* parts I, II, and III only.

3. Build and simulate all the chips specified in the projects/01 directory.