

Chapter 5

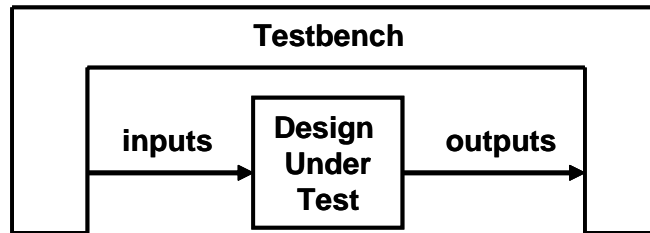
Connecting the Testbench and Design

5.1 Introduction

There are several steps needed to verify a design: generate stimulus, capture responses, determine correctness, and measure progress. But first, you need the proper testbench, connected to the design.

Your testbench wraps around the design, sending in stimulus and capturing the design's response. The testbench forms the "real world" around the design, mimicking the entire environment. For example, a processor model needs to connect to various busses and devices, which are modeled in the testbench as bus functional models. A networking device connects to multiple input and output data streams that are modeled based on standard protocols. A video chip connects to buses that send in commands, and then forms images that are written into memory models. The key concept is that the testbench simulates everything not in the design under test.

Figure 5-1 The testbench – design environment



Your testbench needs a higher-level way to communicate with the design than Verilog's ports and the error-prone pages of connections. You need a robust way to describe the timing so that synchronous signals are always driven and sampled at the correct time and all interactions are free of the race conditions so common to Verilog models.

5.2 Separating the Testbench and Design

In an ideal world, all projects have two separate groups: one to create the design and one to verify it. In the real world, limited budgets may require you to wear both hats. Each team has its own set of specialized skills, such as creating synthesizable RTL code, or figuring out new ways to find bugs in the design. These two groups each read the original design specification and make their own interpretations. The designer has to create code that meets that