This chapter discusses the process of integrating completed macros into the whole chip environment. The topics are:

- Integration overview
- Integrating soft macros
- Integrating hard macros
- Integrating RAMs and datapath generators
- Physical design

10.1 Integration Overview

Chapter 2 described system design from specification to the point where individual blocks could be designed. The succeeding chapters described how these blocks should be designed in order to make them reusable. We now return to the issue of system design, and discuss how to assemble these blocks into the final chip.

At this point in system design, there are two key tasks remaining: physical design and functional verification. Each of these tasks has a dominant challenge. For physical design it is achieving timing closure; for verification, it is knowing when we are done, when we are confident enough in the functionality of the chip that we can tape out and go to fabrication.

In this chapter, we address the integration of the blocks and the physical design of the chip. In the next chapter, we discuss functional verification.
The process of integrating the blocks and doing the physical design can be broken into the following steps:

- Selecting IP blocks and preparing them for integration
- Integrating all the blocks into the top-level RTL
- Planning the physical design
- Synthesis and initial timing analysis
- Initial physical design and timing analysis, with iteration until timing closure
- Final physical design, timing verification, and power analysis
- Physical verification of the design

10.2 Integrating Macros into an SoC Design

Integrating macros into the top-level SoC design poses several challenges. In this section, we will discuss typical integration problems and strategies for dealing with them.

10.2.1 Problems in Integrating IP

Assembling a set of blocks into a top-level design presents a series of challenges to the design team. Naturally, we did a good job of decomposing the design into well-specified blocks, then selected the IP we needed and designed the new blocks required as specified. Nonetheless, when we get down to assembling these blocks and making them work together, we often find issues.

For blocks that were designed specifically for this chip, we tend to find:

- The low level interfaces do not work; for example, a handshake signal is inverted.
- There was a misunderstanding of the functionality of the block.
- There are functional bugs in the design.

Usually we have access to the block designers and the system architect, so these problems are reasonably easy to fix.

For IP that has been obtained from an external source, either a third party or some other division of the company, there are additional problems that frequently occur:

- Someone on the team needs to become familiar enough with the IP to integrate it into the design.
- The documentation is incomplete, making this understanding harder to obtain.
- The interface of the IP does not match the interface of the system bus.