CHAPTER 6

Macro Synthesis Guidelines

This chapter discusses strategies for developing macro synthesis scripts that enable the integrator to synthesize the macro and meet timing goals. The topics include:

- Overview of the synthesis problem
- Synthesis strategies for reusable macros
- High-performance synthesis
- RAM and datapath generators
- Coding guidelines for synthesis scripts

6.1 Overview of the Synthesis Problem

There are some special problems associated with the synthesis of parameterizable soft macros:

- The macro and synthesis scripts must allow the integrator to synthesize the macro and meet timing goals in the final chip.
- The macro must meet timing with the integrator's gate array or standard cell library.
- The macro must meet timing in the integrator's specific configuration of the macro.

This chapter presents a set of tools and methodologies for achieving these goals.
The synthesis guidelines in this chapter are based on many of the same fundamental principles guiding the previous chapter. First and foremost, synthesis and timing design must start at the beginning of the macro design cycle.

That is:

- Functional specifications for the macro must describe the timing, area, wire load model, and power requirements for the design.
- Detailed technical specifications for the macro and its various subblocks must describe the timing requirements and interfaces in detail, including specifications for input and output delays.
- RTL needs to be coded from the outset to meet both the functional and the timing requirements of the design. Coding for functionality first, and then fixing timing problems later, causes significant delays and poor overall performance in many designs.

If these fundamental guidelines are followed, then synthesis is a straightforward issue. Each synthesizable unit or module in the design has a timing budget. Once each module meets this timing budget, the macro is ensured of meeting its overall timing goals. Synthesis problems become localized, so the difficult problems can be solved on small modules, where they are the most tractable.

6.2 Macro Synthesis Strategy

The recommended synthesis strategy for macros is to develop a set of constraints for the macro early in the design process and to use a bottom-up synthesis strategy.

6.2.1 Macro Timing Budget

Rule – The basic timing budget for the macro must be developed as part of the specification process, before the design is partitioned into blocks and before coding begins. This timing budget must be reviewed regularly during the design process to ensure that it is still reasonable and consistent.

The macro timing budget must specify:

- Clock definition
- Setup time requirements for all signals going into the macro
- Clock to output delay requirements for all synchronous outputs of the macro
- Input and output delays for all combinational paths through the macro
- Loading budget for outputs and driving cell for inputs