This chapter discusses system-level issues such as layout, clocking, floorplanning, on-chip busing, and strategies for synthesis, verification, and testing. These elements must be agreed upon before the components of the chip are selected or designed.

Topics in this chapter include:

- Interoperability issues
- Timing and synthesis issues
- Functional design issues
- Physical design issues
- Verification strategy
- Manufacturing test strategies

3.1 Interoperability Issues

Two of the biggest issues affecting the success of a SoC design are the interoperability of the macros being integrated and the interoperability of the tools used to implement the design.

It is essential that the design team agree to a set of design rules for the entire chip before beginning any design or macro selection. By agreeing in advance on such critical issues as clocking and reset strategy, macro interface architecture, and design for test, the team can select and design macros that will work well together.
This chapter describes the basic set of design guidelines that should be explicitly agreed to by the design team before starting implementation or macro selection. Most of these requirements are aimed at facilitating rapid chip-level integration and verification of the various macros.

3.2 Timing and Synthesis Issues

Timing and synthesis issues include synchronous or asynchronous design, clock and reset schemes, and selection of synthesis strategy.

3.2.1 Synchronous vs. Asynchronous Design Style

Rule – The system should be synchronous and register based. Latches should be used only to implement small memories or FIFOs. The FIFOs and memories should be designed so that they are synchronous to the external world and are edge triggered. Exceptions to this rule should be made with great care and must be fully documented.

In the past, latch-based designs have been popular, especially for some processor designs. Multi-phase, non-overlapping clocks were used to clock the various pipeline stages. Latches were viewed as offering greater density and higher performance than register (flop) based designs. These benefits were sufficient to justify the added complexity of design.

Today, the tradeoffs are quite different. Deep submicron technology has made a huge number of gates available to the chip designer and, in most processor-based designs, the size of on-chip memory is dwarfing the size of the processor pipeline. Also, with deep submicron design, delays are dominated by interconnect delay, so the difference in effective delay between latches and flip-flops is minimal.

On the other hand, the cost of the increased complexity of latch-based design has risen significantly with the increase in design size and the need for design reuse.

Latch timing is inherently ambiguous, as illustrated in Figure 3-1. The designer may intend data to be set up at the D input of the latch before the leading edge of the clock, in which case data is propagated to the output on the leading edge of clock. Or, the designer may intend data to be set up just before the trailing edge of the clock, in which case data is propagated to the output (effectively) on the trailing edge of the clock.